CA Process Automation

Referenzhandbuch für Inhaltsdesign

Version 04.2.00

Diese Dokumentation, die eingebettete Hilfesysteme und elektronisch verteilte Materialien beinhaltet (im Folgenden als "Dokumentation" bezeichnet), dient ausschließlich zu Informationszwecken des Nutzers und kann von CA jederzeit geändert oder zurückgenommen werden. Diese Dokumentation ist Eigentum von CA und darf ohne vorherige schriftliche Genehmigung von CA weder vollständig noch auszugsweise kopiert, übertragen, vervielfältigt, veröffentlicht, geändert oder dupliziert werden.

Der Benutzer, der über eine Lizenz für das bzw. die in dieser Dokumentation berücksichtigten Software-Produkt(e) verfügt, ist berechtigt, eine angemessene Anzahl an Kopien dieser Dokumentation zum eigenen innerbetrieblichen Gebrauch im Zusammenhang mit der betreffenden Software auszudrucken, vorausgesetzt, dass jedes Exemplar diesen Urheberrechtsvermerk und sonstige Hinweise von CA enthält.

Dieses Recht zum Drucken oder anderweitigen Anfertigen einer Kopie der Dokumentation beschränkt sich auf den Zeitraum der vollen Wirksamkeit der Produktlizenz. Sollte die Lizenz aus irgendeinem Grund enden, bestätigt der Lizenznehmer gegenüber CA schriftlich, dass alle Kopien oder Teilkopien der Dokumentation an CA zurückgegeben oder vernichtet worden sind.

SOWEIT NACH ANWENDBAREM RECHT ERLAUBT, STELLT CA DIESE DOKUMENTATION IM VORLIEGENDEN ZUSTAND OHNE JEGLICHE GEWÄHRLEISTUNG ZUR VERFÜGUNG; DAZU GEHÖREN INSBESONDERE STILLSCHWEIGENDE GEWÄHRLEISTUNGEN DER MARKTTAUGLICHKEIT, DER EIGNUNG FÜR EINEN BESTIMMTEN ZWECK UND DER NICHTVERLETZUNG VON RECHTEN. IN KEINEM FALL HAFTET CA GEGENÜBER IHNEN ODER DRITTEN GEGENÜBER FÜR VERLUSTE ODER UNMITTELBARE ODER MITTELBARE SCHÄDEN, DIE AUS DER NUTZUNG DIESER DOKUMENTATION ENTSTEHEN; DAZU GEHÖREN INSBESONDERE ENTGANGENE GEWINNE, VERLORENGEGANGENE INVESTITIONEN, BETRIEBSUNTERBRECHUNG, VERLUST VON GOODWILL ODER DATENVERLUST, SELBST WENN CA ÜBER DIE MÖGLICHKEIT DIESES VERLUSTES ODER SCHADENS INFORMIERT WURDE.

Die Verwendung aller in der Dokumentation aufgeführten Software-Produkte unterliegt den entsprechenden Lizenzvereinbarungen, und diese werden durch die Bedingungen dieser rechtlichen Hinweise in keiner Weise verändert.

Diese Dokumentation wurde von CA hergestellt.

Zur Verfügung gestellt mit „Restricted Rights“ (eingeschränkten Rechten) geliefert. Die Verwendung, Duplizierung oder Veröffentlichung durch die US-Regierung unterliegt den in FAR, Absätze 12.212, 52.227-14 und 52.227-19(c)(1) bis (2) und DFARS, Absatz 252.227-7014(b)(3) festgelegten Einschränkungen, soweit anwendbar, oder deren Nachfolgebestimmungen.

Copyright © 2013 CA. Alle Rechte vorbehalten. Alle Markenzeichen, Markennamen, Dienstleistungsmarken und Logos, auf die hier verwiesen wird, sind Eigentum der jeweiligen Unternehmen.

CA Technologies-Produktreferenzen

Dieses Dokument bezieht sich auf die folgenden Produkte von CA Technologies:

- CA Catalyst für CA Service Desk Manager (CA Catalyst-Connector für CA SDM)
- CA Client Automation (früher CA IT Client Manager)
- CA Configuration Automation (früher CA Cohesion®
  Application Configuration Manager)
- CA Configuration Management Database (CA CMDB)
- CA eHealth®
- CA Embedded Entitlements Manager (CA EEM)
- CA Infrastructure Insight (früher im Paket: CA Spectrum IM und
  CA NetQoS Reporter Analysator kombiniert)
- CA NSM
- CA Process Automation (früher CA IT Process Automation Manager)
- CA Service Catalog
- CA Service Desk Manager (CA SDM)
- CA Service Operations Insight (CA SOI) (früher CA Spectrum® Service Assurance)
- CA SiteMinder®
- CA Workload Automation AE

Technischer Support – Kontaktinformationen

Wenn Sie technische Unterstützung für dieses Produkt benötigen, wenden Sie sich an den Technischen Support unter http://www.ca.com/worldwide. Dort finden Sie eine Liste mit Standorten und Telefonnummern sowie Informationen zu den Bürozeiten.

Änderungen in der Dokumentation

Die folgenden Dokumentationsaktualisierungen sind seit der letzten Ausgabe dieser Dokumentation vorgenommen worden:

- [Topic Name]—[Description of new chapter, topic, or description of changes to an existing topic].
- [Topic Name]—[Description of new chapter, topic, or description of changes to an existing topic].

Inhalt

Kapitel 1: Einführung in Operatoren 19

Operatorübersicht 19
Ausführungsorte von Operatoren 21
Allgemeine Eigenschaften aller Operatoren 26
    Ausführungseinstellungen (Alle Operatoren) 27
Allgemeine Eigenschaften von Operatoren in Prozessen 33
    Verarbeitungsgruppe 33
    Simulationseigenschaften 35
    Informationseigenschaften 38
Allgemeine Eigenschaften von Operatoren in Ablaufplänen 40
    Kalender-Einstellungen 40
    Manuell eingeschlossene Daten 41
    Manuell ausgeschlossene Daten 42
    Name der Aufgabe 42
Gewöhnliche Ausgabeparameter für alle Operatoren 42

Kapitel 2: Standardoperatoren 45

Operator "Start" 45
    Eingabeparameter 45
Operator "Kommentar" 46
    Eingabeparameter 46
Operator "Bei Erfolg beenden" 47
    Eingabeparameter 48
Operator "Bei Fehlschlag beenden" 49
    Eingabeparameter 49
Operator "Prozessfortschritt" 50
    Eingabeparameter 51
    Ausgabe 51
Operator "Und" 51
Operator "Oder" 52
Operator "Zurücksetzen" 52
    Eingabeparameter 53
    So funktioniert der Operator "Zurücksetzen" mit dem Operator "Schleife" 54
Schleifenoperator 54
    Eingabeparameter 55
    Ausgabeparameter 58

Manuelles Zurücksetzen des Operators "Schleife" in einem Prozess 58
Operator "Prozessspur ändern" 58
Eingabeparameter 59
Operator "Ausnahme" 59
Eingabeparameter 59
Links 60
Verknüpfungseigenschaften 60

Kapitel 3: Catalyst 61

Generische USM-Operatoren 61
Operator "Erstellen" 61
Operator "Löschen" 63
Operator "Ausführen" 64
Operator "Abrufen" 71
Operator "Änderungen abonnieren" 72
Nicht generische USM-Operatoren 76
Catalyst-Sicherheitsparameter 79

Kapitel 4: Befehls-Ausführung 81

Operator "Programm ausführen" 81
Eingabeparameter 82
Ausgabeparameter 85
Operator "Skript ausführen" 86
Eingabeparameter 86
Ausgabeparameter 92
Operator "SSH-Befehl ausführen" 93
Eingabeparameter 93
Ausgabeparameter 102
Operator-Ports 104
Beispiel 106
Operator "SSH-Skript ausführen" 107
Eingabeparameter 109
Ausgabeparameter 119
Operator-Ports 120
Beispiel 122
Operator "Telnet-Befehl ausführen" 126
Eingabeparameter 126
Ausgabeparameter 134
Operator-Ports 137
Beispiel 139
Operator "Telnet-Skript ausführen" 141

Eingabeparameter 142
Ausgabeparameter 148
Operator-Ports 149
Beispiel 150

Kapitel 5: Datenbanken 155

Oracle-Parameter 155
MSSQL-Server-Parameter 157
Parameter von MySQL 158
Sybase-Parameter 158
Eigenschaften auf Operatorebene 159
Anmeldeparameter für den Datenbankserver 160
Operator "Masseneinfügung in Datenbank" 165
Eingabeparameter 165
Ausgabeparameter 166
Operator "Aus Datenbank löschen" 166
Eingabeparameter 167
Ausgabeparameter 167
Operator "Datenbankschema abrufen" 169
Eingabeparameter 169
Ausgabeparameter 169
Operator "Freien Speicherplatz abrufen" 170
Eingabeparameter 170
Ausgabeparameter 170
Operator "Gespeicherte Prozedur abrufen" 171
Eingabeparameter 172
Ausgabeparameter 172
Operator "Tabelle abrufen" 173
Eingabeparameter 174
Ausgabeparameter 174
Operator "Verwendeten Speicherplatz abrufen" 175
Eingabeparameter 176
Ausgabeparameter 176
Operator "Version abrufen" 177
Eingabeparameter 178
Ausgabeparameter 178
Operator "Ansicht abrufen" 178
Eingabeparameter 179
Ausgabeparameter 179
Operator "In Datenbank einfügen" 180
Eingabeparameter 181

Ausgabeparameter 181
Operator "Datenbank abfragen" 183
Eingabeparameter 183
Ausgabeparameter 184
Ausführen einer gespeicherten Prozedur 187
Operator "Aus Datenbank auswählen" 189
Eingabeparameter 189
Ausgabeparameter 191
Operator "In Datenbank aktualisieren" 192
Eingabeparameter 192
Ausgabeparameter 193

Kapitel 6: Datum - Uhrzeit 195

Operator "Kalender überprüfen" 195
Eingabeparameter 195
Ausgabeparameter 197
Operator "Datum/Uhrzeit überprüfen" 197
Eingabeparameter 198
Ausgabeparameter 199

Kapitel 7: Verzeichnisdienste 201

LDAP-Anmeldeparameter 201
Operator "Computer zu Domäne hinzufügen" 202
Eingabeparameter 202
Ausgabeparameter 202
Beispiel 202
Operator-Fehler 203
Operator "Anwender zur Gruppe hinzufügen" 203
Eingabeparameter 203
Ausgabeparameter 204
Beispiel 204
Operator-Fehler 204
Operator "Gruppe erstellen" 205
Eingabeparameter 205
Ausgabeparameter 207
Beispiel 208
Operator-Fehler 208
Operator "Objekt erstellen" 209
Eingabeparameter 209
Ausgabeparameter 212
Beispiel 212

Operator-Fehler 213
Operator "Organisationseinheit erstellen" 214
Eingabeparameter 214
Ausgabeparameter 214
Beispiel 214
Operator-Fehler 215
Operator "Anwender erstellen" 215
Eingabeparameter 215
Ausgabeparameter 218
Beispiel 219
Operator-Fehler 219
Operator "Objekt löschen" 220
Eingabeparameter 220
Ausgabeparameter 221
Beispiele 222
Operator "Domänencontroller abrufen" 227
Eingabeparameter 227
Ausgabeparameter 228
Beispiel 229
Operator-Fehler 229
Operator "Ruhendes Konto abrufen" 230
Eingabeparameter 230
Ausgabeparameter 231
Operator "Objekt abrufen" 233
Eingabeparameter 233
Ausgabeparameter 240
Beispiele 242
Operator-Fehler 248
Operator "Anwender abrufen" 248
Eingabeparameter 248
Ausgabeparameter 251
Beispiele 253
Operator-Fehler 255
Operator "Objekt verschieben" 256
Eingabeparameter 256
Ausgabeparameter 256
Beispiel 256
Operator-Fehler 257
Operator "Anwender aus Gruppe entfernen" 257
Eingabeparameter 257
Ausgabeparameter 258
Beispiel 258

Operator-Fehler 258
Operator "Objektattribute aktualisieren" 259
Eingabeparameter 259
Ausgabeparameter 263
Operator-Fehler 264
Operator "Anwenderstartverzeichnis aktualisieren" 264
Eingabeparameter 264
Ausgabeparameter 265
Beispiel 265
Operator-Fehler 265
Hinzufügen eines SSL-Zertifikats zu CA Process Automation 266

Kapitel 8: E-Mail 269

Allgemeine Parameter für E-Mail-Operatoren 269
Filterkriterien für Meldungen 270
Anmeldeparameter für den Mail-Server 272
Operator "Ordner erstellen" 273
Eingabeparameter 273
Ausgabeparameter 273
Operator "E-Mail löschen" 274
Eingabeparameter 274
Ausgabeparameter 275
Operator "Ordner löschen" 275
Eingabeparameter 275
Ausgabeparameter 276
Beispiel 276
Operator "E-Mail-Inhalt abrufen" 277
Eingabeparameter 277
Ausgabeparameter 279
Operator "E-Mail-Anzahl abrufen" 280
Eingabeparameter 280
Ausgabeparameter 281
Operator "E-Mail-Umschlag abrufen" 281
Eingabeparameter 282
Ausgabeparameter 283
Operator "E-Mail-Liste abrufen" 284
Eingabeparameter 284
Ausgabeparameter 284
Operator "E-Mail verschieben" 286
Eingabeparameter 286
Ausgabeparameter 287

Operator "Ordner bereinigen" 287
Eingabeparameter 288
Ausgabeparameter 288
Operator "Ordner umbenennen" 288
Eingabeparameter 288
Ausgabeparameter 289
Operator "E-Mail senden" 289
Eingabeparameter 290
Ausgabeparameter 291

Kapitel 9: Dateimanagement 293

Operator "Datei komprimieren" 293
Eingabeparameter 294
Ausgabeparameter 294
Operator "Datei kopieren" 295
Eingabeparameter 295
Ausgabeparameter 296
Operator "Ordner erstellen" 296
Eingabeparameter 296
Ausgabeparameter 297
Operator "Datei dekomprimieren" 297
Eingabeparameter 297
Ausgabeparameter 298
Operator "Datei löschen" 298
Eingabeparameter 298
Ausgabeparameter 299
Operator "Verzeichnisinhalt abrufen" 299
Eingabeparameter 299
Ausgabeparameter 301
Operator "Dateiattribute abrufen" 302
Eingabeparameter 302
Ausgabeparameter 303
Beispiel 304
Operator "Datei überwachen" 305
Eingabeparameter 306
Ausgabeparameter 308
Beispiel 308
Operator "Lesen ab Datei" 309
Eingabeparameter 309
Ausgabeparameter 310
Beispiel (Operator "Lesen aus Datei") 311

Operator "Datei umbenennen" 314
Eingabeparameter 314
Ausgabeparameter 314
Operator "Dateiinhalt durchsuchen" 315
Eingabeparameter 315
Ausgabeparameter 317
Operator "Dateieigentümer aktualisieren" 318
Eingabeparameter 318
Ausgabeparameter 319
Operator "Dateiberechtigung aktualisieren" 319
Eingabeparameter 319
Ausgabeparameter 320
Operator "Dateizeitstempel aktualisieren" 320
Eingabeparameter 320
Ausgabeparameter 322
Operator "Datei schreiben" 322
Eingabeparameter 322
Ausgabeparameter 323

Kapitel 10: Dateiübertragung 325

Operator "Verzeichnis erstellen" 325
Eingabeparameter 326
Ausgabeparameter 327
Operator "Verzeichnis löschen" 327
Eigenschaften von "Remote-Verzeichnis löschen" 327
Ausgabeparameter 329
Operator "Datei löschen" 329
Eingabeparameter 329
Ausgabeparameter 330
Operator "Datei herunterladen" 331
Eigenschaften von "Remote-Datei abrufen" 331
Ausgabeparameter 333
Operator "Dateiinformationen abrufen" 333
Eingabeparameter 334
Ausgabeparameter 336
Operator "Datei verschieben" 337
Eingabeparameter 337
Ausgabeparameter 339
Operator "Datei über TFTP herunterladen" 339
Eingabeparameter 340
Ausgabeparameter 341

Operator-Ports 341
Operator "Datei über TFTP hochladen" 342
Eingabeparameter 342
Ausgabeparameter 343
Operator-Ports 344
Operator "Datei laden" 345
Eingabeparameter 345
Ausgabeparameter 347

Kapitel 11: Java-Verwaltung 349

JMX-Anmeldeparameter 349
Operator "MBean-Attribute abrufen" 350
Eingabeparameter 350
Ausgabeparameter 351
Beispiel 352
Operator "MBean-Methode aufrufen" 353
Eingabeparameter 353
Ausgabeparameter 354
Beispiel 355
Operator "MBean-Attribute aktualisieren" 356
Eingabeparameter 356
Ausgabeparameter 357
Beispiel 358

Kapitel 12: Netzwerkhilfsprogramme 361

Operator "Lokale Netzwerkschnittstellen abrufen" 361
Eingabeparameter 362
Ausgabeparameter 362
Operator-Ports 363
Beispiel 364
Operator "Status des Netzwerkdienstes abrufen" 364
Eingabeparameter 365
Ausgabeparameter 368
Operator-Ports 370
Operator "SNMP-Variable abrufen" 372
Eingabeparameter 373
Ausgabeparameter 374
Operator "SNMP-Variable überwachen" 375
Eingabeparameter 375
Ausgabeparameter 378
Operator "Host pingen" 379

Eingabeparameter 379
Ausgabeparameter 381
Operator-Ports 383
Operator "SNMP-Trap senden" 383
Eingabeparameter 384
Ausgabeparameter 385
Operator "SNMP-Variable aktualisieren" 386
Eingabeparameter 386
Ausgabeparameter 387

Kapitel 13: Prozesssteuerung 389

Operator "Anwenderaufgabe zuweisen" 389
Eingabeparameter 391
Ausgabeparameter 395
Beispiel 395
Operator "Ausdruck auswerten" 398
Eingabeparameter 398
Ausgabeparameter 399
Operator "Ressourcen verwalten" 399
Eingabeparameter 400
Ausgabeparameter 403
Event-Operatoren 403
Operator "Event überwachen" 404
Operator "Event senden" 406
Verwendungsmuster für Events 407
Operator "Prozess starten" 408
Eingabeparameter 408
Ausgabeparameter 410

Kapitel 14: Hilfsprogramme 411

Operator "XPath anwenden" 411
Eingabeparameter 412
Ausgabeparameter 413
Anwenden von XSLT-Operatoren 413
Eingabeparameter 414
Ausgabeparameter 415
Operator "Verzögerung" 416
Eingabeparameter 416
Ausgabeparameter 417
Operator "Java aufrufen" 418
Eingabeparameter 418

Ausgabeparameter 431
Java-Beispiel 434
Ressource für die Ausführung des Operators "Java aufrufen" - Beispiel 436
Operator-Ports 436
Operator "JavaScript ausführen" 438
Eingabeparameter 438
Ausgabeparameter 438

Kapitel 15: Webservices 439

HTTP-Operatoren: Allgemeine Eingabeparameter 439
Informationen zu HTTP-URL 440
HTTP-Proxy-Informationen 444
Informationen zu HTTP-Header 448
Informationen zu HTTP-Cookies 449
Informationen zum Inhalt der HTTP-Antwort 449
Informationen zur HTTP-Konfiguration 451
HTTP-Operatoren: Allgemeine Ausgabeparameter 453
HTTP-Operatoren: Allgemeine Ausgabe-Ports 458
Operator "HTTP DELETE" 460
Eingabeparameter 461
Ausgabeparameter 462
Operator "HTTP GET" 463
Eingabeparameter 464
Ausgabeparameter 466
Operator-Ausfall 467
Operator "HTTP HEAD" 468
Eingabeparameter 468
Ausgabeparameter 471
Operator-Ausfall 473
Operator "HTTP OPTIONS" 473
Eingabeparameter 473
Ausgabeparameter 475
Operator "HTTP POST" 477
Eingabeparameter 478
Ausgabeparameter 481
Operator-Ausfall 483
Operator "HTTP POST FORM" 483
Eingabeparameter 484
Ausgabeparameter 489
Operator-Ausfall 491
Operator "HTTP PUT" 491

Eingabeparameter 492
Ausgabeparameter 495
Operator "HTTP TRACE" 497
Eingabeparameter 497
Ausgabeparameter 499
Operator "SOAP-Methode aufrufen" 500
Eingabeparameter 501
Ausgabeparameter 522
Operator "Asynchrone SOAP-Methode aufrufen" 527
Eingabeparameter 527
Ausgabeparameter 548

Kapitel 16: Systemfunktionen von CA Process Automation 555

Rückgabetypen 555
absPath 556
adjustDate 557
adjustResourceVals 558
applyXPath 559
applyXPathToUrl 561
checkCalendarDate 563
convertJson 564
convertValueToXml 566
convertXml 567
convertXmlUrl 568
createHyperLink 568
createResourceObject 569
deleteAttachments 569
deleteObject 570
deleteResource 571
deleteValueMapField 571
existsCalendar 572
existsCustomIcon 573
existsCustomOperator 573
existsDataset 574
existsFolder 575
existsInteractionRequestForm 576
existsProcess 576
existsProcessWatch 577
existsResource 578
existsSchedule 578
formatDate 579

formatString 580
getAllAttachments 581
getAttachmentContent 581
getCountOfProcessStates 582
getEEMArtifactToken 582
getEEMArtifactTokenForUser 583
getEEMCredentialsToken 584
getEEMCredentialsTokenForUser 585
getEnvVar 586
getOrchestratorURL 586
getPartialAttachmentContent 587
getResourceAvail 587
getResourceName 588
getResourceTotal 588
getTouchpoints 589
getValueFromValueMapArray() 590
getValueMapFields 591
getValuesFromValueMapArray() 592
hasField 592
include 593
isFIPSMode 594
isTouchpointUp 594
load 595
lockResource 596
logEvent 597
newValueMap 598
newDataset 598
nextOpenDate 599
now 599
parseDate 600
resetResource 601
rollDate 601
rollTime 602
saveAttachmentToFile 603
setOperatorStatus 604
setProcessProgress 605
setResourceTotal 605
today 606

Kapitel 1: Einführung in Operatoren

Diese Referenz enthält Informationen zu den Operatoren, die in CA Process Automation enthalten sind. Operatoren sind in Kategorien gruppiert. Dieses Handbuch gruppiert die Beschreibungen von Operatorinformationen nach diesen Kategorien.

Das *Referenzhandbuch für Inhaltsdesign* beschreibt auch Systemfunktionen. Verwenden Sie die Systemfunktionen, um anwenderspezifische JavaScripts zu schreiben. Diese JavaScripts können in Operatoren platziert werden, um die in CA Process Automation verwendeten Daten zu ändern.

Operatorübersicht

Container für Operatorkategorien werden als Ordner im Auswahlmenü "Operatoren" im Designer angezeigt.

CA Process Automation enthält die folgenden Operatorkategorien:

Standard (siehe Seite 45)

Standardmäßige Operatoren schließen wesentliche Funktionalitätsoperatoren ein, die Workflows in Prozessen steuern. Grundlegende Funktionen wie Starten, Anhalten, Verknüpfen und Kommentieren werden mit den standardmäßigen Operatoren bereitgestellt. Sie können auch Optionen für Schleifen und zum Zurücksetzen festlegen und Prozessspurwechsel mithilfe dieser Operatoren einbinden.

Catalyst (siehe Seite 61)

Catalyst-Operatoren unterstützen die UCF-Schnittstellen zum Erstellen, Lesen, Aktualisieren und Löschen (create, read, update, delete - CRUD) sowie für Event-Abonnements. Die Operatoren zeigen Objekttypen und Eigenschaften des vereinheitlichten Servicemodells (USM) an.

Befehls-Ausführung (siehe Seite 81)

Das Befehlsausführungsmodul führt Prozesse und Skripte in der Host-Betriebsumgebung aus.

Datenbanken (siehe Seite 155)

Datenbanken-Operatoren stellen eine Möglichkeit bereit, um zu kommunizieren und Datenbankabfragen gegen unterschiedliche Datenbankserver auszuführen.

Datum - Uhrzeit (siehe Seite 195)

Datum/Uhrzeit-Operatoren verwalten das Datum und die Uhrzeit des CA Process Automation-Servers.

Verzeichnisdienste (siehe Seite 201)

Directory-Service-Operatoren unterstützen das Lightweight Directory Access Protocol. Mit Ausnahme der Active Directory-spezifischen Operatoren arbeiten alle Operatoren mit verschiedenen LDAP-Servern.

E-Mail (siehe Seite 269)

E-Mail-Operatoren automatisieren auf E-Mails und Ordner in einem E-Mail-Server ausgeführte Aufgaben. E-Mail-Operatoren lesen E-Mails vom Mail-Server durch IMAP/POP3.

Dateimanagement (siehe Seite 293)

Dateimanagement-Operatoren überwachen Verzeichnisse, Dateien und ihre Inhalte. Dateimanagementoperatoren können entweder lokal oder auf einem Remote-System ausgeführt werden. Diese Operatoren können lokale Dateien erstellen, löschen, umbenennen, komprimieren und dekomprimieren und Dateien auf dem Kontaktpunkt, auf dem die Dateimanagement-Kategorie ausgeführt wird, überwachen.

Dateiübertragung (siehe Seite 325)

Dateiübertragungs-Operatoren ermöglichen Ihnen, FTP und SFTP zu verwenden.

Java-Verwaltung (siehe Seite 349)

Java-Verwaltungsoperatoren bieten eine Verwaltungsschnittstelle für Systeme, die JMX unterstützen.

Netzwerkhilfsprogramme (siehe Seite 361)

Netzwerkhilfsprogramme-Operatoren ermöglichen es dem Anwender, mit anderen Netzwerkgeräten über SNMP zu kommunizieren.

Prozesssteuerung (siehe Seite 389)

Prozesssteuerungs-Operatoren können CA Process Automation-Prozesse ausführen, überwachen und steuern.

Hilfsprogramme (siehe Seite 411)

Hilfsprogramme-Operatoren rufen externe JAR-Dateien in CA Process Automation auf.

Webservices (siehe Seite 439)

Webservices-Operatoren stellen verschiedene standardmäßige Netzwerkprotokollhilfsprogramme für die automatischen, von CA Process Automation ermöglichten Business-Prozesse bereit.

Ausführungsorte von Operatoren

Ein Operator wird auf einem Koordinationsrechner standardmäßig ausgeführt. Sie können jedoch einen spezifischen Speicherort (Ziel) in den Ausführungseinstellungen angeben.

Ziele können als ein Kontaktpunkt, Kontaktpunktgruppe, Agent-ID, Proxy-Kontaktpunkt oder eine IP-Adresse oder FQDN angegeben werden. Eine Beschreibung von allen diesen Elementen finden Sie im *Handbuch für Inhaltsdesign* unter "Angabemöglichkeiten von Zielen für einen Operatoren". Jedes Ziel wird letztlich zu einem Koordinationsrechner, zu mindestens einem Agent oder einem Remote-Host mit einer SSH-Verbindung zu einem Agenten aufgelöst. Der Ort, an dem der Operator ausgeführt wird, ist wichtig, da der entworfene Prozess in einigen Fällen auf einem spezifischen Host ausgeführt werden muss. Viele Operatoren können entweder auf einem Koordinationsrechner oder auf Agenten ausgeführt werden. Allerdings können bestimmte Operatoren nur auf Koordinationsrechnern ausgeführt werden, und andere Operatoren können nur auf Agenten ausgeführt werden. Für einige Operatoren ist es erforderlich, dass der abgezielte Remote-Host den SSH-Zugriff unterstützt.

Wir empfehlen, dass Sie einen Agenten zur Ausführung des Operators "Java aufrufen" verwenden. Dies ist der einzige CA Process Automation-Operator, der nicht auf einem Koordinationsrechner ausgeführt werden kann.

Hinweis: Als Proxy-Kontaktpunkte fungierende Agenten haben unterschiedliche Kategorieanforderungen. Vorgänge, die Proxy-Kontaktpunkte (oder einen Remote-Host, der von einer Hostgruppe referenziert wird) als Ziel festlegen, laufen durch SSH.

Kategorie	Operator	Koordinationsrechner	Agent	Remote-Host (SSH-Ziel)
Catalyst (siehe Seite 61)	Alle	x		
Befehls-Ausführung (siehe Seite 81)				
	Programm ausführen	x	x	x
	SSH-Befehl ausführen	x	x	
	SSH-Skript ausführen	x	x	
	Skript ausführen	x	x	x
	Telnet-Befehl ausführen	x	x	
	Telnet-Skript ausführen	x	x	
Datenbanken (siehe Seite 155)				
	Masseneinfügung in Datenbank	x	x	x

Kategorie	Operator	Koordinationsrechner	Agent	Remote-Host (SSH-Ziel)
	Aus Datenbank löschen	x	x	x
	Datenbankschema abrufen	x	x	x
	Freien Speicherplatz abrufen	x	x	x
	Gespeicherte Prozedur abrufen	x	x	x
	Tabelle abrufen	x	x	x
	Verwendeten Speicherplatz abrufen	x	x	x
	Version abrufen	x	x	x
	Ansicht abrufen	x	x	x
	In Datenbank einfügen	x	x	x
	Datenbank abfragen	x	x	x
	Aus Datenbank auswählen	x	x	x
	In Datenbank aktualisieren	x	x	x
Datum - Uhrzeit (siehe Seite 195)				
	Kalender überprüfen	x		
	Datum/Uhrzeit überprüfen	x		
Verzeichnisdienste (siehe Seite 201)				
	Computer zu Domäne hinzufügen	x	x	
	Anwender zur Gruppe hinzufügen	x	x	
	Gruppe erstellen	x	x	
	Objekt erstellen	x	x	
	Organisationseinheit erstellen	x	x	
	Anwender erstellen	x	x	
	Objekt löschen	x	x	
	Domänencontroller abrufen	x	x	
	Ruhendes Konto abrufen	x	x	
	Objekt abrufen	x	x	

Kategorie	Operator	Koordinationsrechner	Agent	Remote-Host (SSH-Ziel)
	Anwender abrufen	x	x	
	Objekt verschieben	x	x	
	Anwender aus Gruppe entfernen	x	x	
	Objektattribute aktualisieren	x	x	
	Anwenderstartverzeichnis aktualisieren	x	x	
E-Mail (siehe Seite 269)				
	Ordner erstellen	x	x	
	E-Mail löschen	x	x	
	Ordner löschen	x	x	
	E-Mail-Inhalt abrufen	x	x	
	E-Mail-Anzahl abrufen	x	x	
	E-Mail-Umschlag abrufen	x	x	
	E-Mail-Liste abrufen	x	x	
	E-Mail verschieben	x	x	
	Ordner bereinigen	x	x	
	Ordner umbenennen	x	x	
	E-Mail senden	x	x	
Dateimanagement (siehe Seite 293)				
	Datei komprimieren	x	x	x
	Datei kopieren	x	x	x
	Ordner erstellen	x	x	x
	Datei dekomprimieren	x	x	x
	Datei löschen	x	x	x
	Verzeichnisinhalt abrufen	x	x	x
	Dateiattribute abrufen	x	x	x
	Datei überwachen	x	x	x
	Lesen ab Datei	x	x	x

Kategorie	Operator	Koordinationsrechner	Agent	Remote-Host (SSH-Ziel)
	Datei umbenennen	x	x	x
	Dateiinhalt durchsuchen	x	x	x
	Dateieigentümer aktualisieren	x	x	x
	Dateiberechtigung aktualisieren	x	x	x
	Dateizeitstempel aktualisieren	x	x	x
	Datei schreiben	x	x	x
Dateiübertragung (siehe Seite 325)				
	Verzeichnis erstellen	x	x	x
	Verzeichnis löschen	x	x	x
	Datei löschen	x	x	x
	Datei herunterladen	x	x	x
	Dateiinformationen abrufen	x	x	x
	Datei verschieben	x	x	x
	Datei über TFTP herunterladen	x	x	
	Datei über TFTP hochladen	x	x	
	Datei hochladen	x	x	x
Java-Verwaltung (siehe Seite 349)				
	MBean-Attribute abrufen	x	x	
	MBean-Methode aufrufen	x	x	
	MBean-Attribute aktualisieren	x	x	
Netzwerkhilfsprogramme (siehe Seite 361)				
	Lokale Netzwerkschnittstellen abrufen	x	x	

Kategorie	Operator	Koordinationsrechner	Agent	Remote-Host (SSH-Ziel)
	Status des Netzwerkdienstes abrufen	x	x	
	SNMP-Variable abrufen	x	x	
	SNMP-Variable überwachen	x	x	
	Host pingen	x	x	
	SNMP-Trap senden	x	x	
	SNMP-Variable aktualisieren	x	x	
Prozesssteuerung (siehe Seite 389)				
	Anwenderaufgabe zuweisen	x		
	Ausdruck auswerten	x		
	Ressourcen verwalten	x		
	Event überwachen	x		
	Event senden	x		
	Prozess starten	x		
Standard (siehe Seite 45)				
	Start	x		
	Kommentar	x		
	Bei Erfolg beenden	x		
	Bei Fehlschlag beenden	x		
	Prozessfortschritt	x		
	Und	x		
	ODER	x		
	Zurücksetzen	x		
	Schleife	x		
	Prozessspur ändern	x		
	Ausnahme	x		
Hilfsprogramme (siehe Seite 411)				
	XPath anwenden	x		
	XSLT anwenden	x		

Kategorie	Operator	Koordinationsrechner	Agent	Remote-Host (SSH-Ziel)
	Verzögerung	x		
	Java aufrufen		x	
	JavaScript ausführen	x		
Webservices (siehe Seite 439)				
	HTTP Delete	x	x	
	HTTP Get	x	x	
	HTTP Head	x	x	
	HTTP Options	x	x	
	HTTP Post	x	x	
	HTTP Post Form	x	x	
	HTTP Put	x	x	
	HTTP Trace	x	x	
	SOAP-Methode aufrufen	x	x	
	Asynchrone SOAP-Methode aufrufen	x	x	

Allgemeine Eigenschaften aller Operatoren

Operator-Konfigurationsoptionen werden im Fenster "Eigenschaften" angezeigt, wenn Sie in einem Prozess oder Ablaufplanobjekt auf einen Operator doppelklicken.

Die Konfiguration für Operatorkategorien wird im *Handbuch für Inhaltsadministratoren* beschrieben.

Hinweis: Stellen Sie sicher, dass das Kontrollkästchen "Eigenschaften" aktiviert ist, wenn die Eigenschaften für einen Operator nicht angezeigt werden. Wählen Sie in der oberen rechten Ecke des Designers "Anzeigen", "Eigenschaften", und maximieren Sie anschließend das Fenster "Eigenschaften" im unteren Fensterbereich. Wenn die Operatoren selbst nicht angezeigt werden, klicken Sie unter dem Menü "Anzeigen" auf "Operatoren" .

Ausführungseinstellungen (Alle Operatoren)

Ziel

Gibt das Ziel an, auf dem der Operator ausgeführt werden soll. Ein Ziel kann ein Kontaktpunkt, eine Kontaktpunktgruppe, Agenten-ID, Proxy-Kontaktpunkt, IP-Adresse oder FQDN sein. Stellen Sie sicher, dass Sie die Kategorie für den Operator auf dem Kontaktpunkt aktivieren. Um das Dialogfeld "Objektbrowser" zu öffnen und einen Kontaktpunkt auszuwählen, klicken Sie auf "Auswählen".

"Ziel" ist ein berechneter Ausdruck

Gibt das Ziel mithilfe eines Ausdrucks an. Um ein Ziel zur Laufzeit dynamisch anzugeben, verwenden Sie einen berechneten Ausdruck. Lesen Sie sich folgende Beispiele sorgfältig durch:

- Verwenden Sie eine Zeichenfolgen-Datensatzvariable, die den Namen des Kontaktpunkts enthält.
- Verwenden Sie eine Objektreferenz-Datensatzvariable, die auf den Kontaktpunkt verweist.

Wichtig! Wenn ein Prozess für den Import als vordefinierter Inhalt bestimmt ist, geben Sie die IP-Adresse oder den FQDN in einem Datensatz an. Ein Datensatz kann in der Importumgebung geändert werden, das Feld "Ziel" kann aber nicht geändert werden.

Ziel nur in Hostgruppen abgleichen

Gibt an, wie der Zielname aufgelöst wird.

Ausgewählt

Gibt an, dass das Feld "Ziel" einen Hostnamen oder eine IP-Adresse enthält, auf den oder die von einer Hostgruppe verwiesen wird.

Hinweis: Wenn das Feld "Ziel" einen Kontaktpunkt, einen Proxy-Kontaktpunkt oder eine Agenten-ID enthält, schlägt die Ausführung fehl.

Gelöscht

Gibt an, dass das Feld "Ziel" den Namen eines Kontaktpunkts, einen Proxy-Kontaktpunkt oder eine Agenten-ID enthält.

Hinweis: Wenn das Feld "Ziel" eine gültige Hostgruppen-Referenz enthält, wird die Verarbeitung durchgeführt, aber nicht optimiert.

Schleife

CA Process Automation ermöglicht es Ihnen, einen Operator in einer Schleife auszuführen, bis eine Bedingung erfüllt wird. Die Eigenschaft "Schleife" gibt an, wie häufig ein Operator wiederholt wird.

Die Beendigungsbedingungen und die verbindenden Verknüpfungen vom Operator, der in einer Schleife ausgeführt wird, werden nur ausgewertet, wenn die Schleife beendet wird.

Hinweis: Ausführlichere Informationen zur Verwendung von Schleifen in CA Process Automation finden Sie im *Handbuch für Inhaltsdesign*.

Wiederholungsanzahl

Gibt die Bedingung an, die für die Schleifen erforderlich ist. Zwei Optionen sind verfügbar:

- Sie können eine Ganzzahl oder CA Process Automation-Ausdruck eingeben, der zur Laufzeit eine Ganzzahl zurückgibt. Der Standardwert 1 führt eine Schleife einmal für einen Operator aus. Um eine Endlosschleife auszuführen, klicken Sie auf das Kontrollkästchen "Endlosschleife".
- Geben Sie einen booleschen Ausdruck an. Der Ausdruck wird ausgewertet, nachdem der Operator ausgeführt wurde. Wenn der Ausdruck als Wahr ausgewertet wird, wird die Schleife fortgesetzt. Wenn der Ausdruck als Falsch ausgewertet wird, wird die Schleife abgeschlossen.

Endlosschleife

Wenn diese Option ausgewählt ist, wird "Wiederholungsanzahl" ignoriert und eine Endlosschleife erzeugt. Der Operator wird so lange wiederholt, bis eine der beiden Bedingungen eintritt:

- Der Prozess wird unterbrochen.
- Die Schleife wird von einer anderen Verzweigung des Prozesses angehalten (indem eine Befehlsverknüpfung zur Beendigung einer Schleife für den Operator "Schleife" (siehe Seite 54) ausgeführt wird).

Verzögerung zwischen Iterationen

Gibt die Verzögerung zwischen zwei Schleifeniterationen in Sekunden an (der Standardwert ist 0).

Zeitlimit

Ermöglicht es Ihnen, ein Zeitlimit als Teil jedes Operators festzulegen. Wenn der Operator nicht innerhalb der angegebenen Zeit abgeschlossen wird, gibt diese Einstellung eine Beendigungsstrategie an. Die Option "Zeitlimit" bietet eine der folgenden Möglichkeiten:

- Den Operator beenden und den alternativen Zeitlimit-Pfad verwenden.
- Mit dem Operator fortfahren während der Zeitlimit-Pfad verwendet wird.
- Den Operator zurücksetzen (den Operator erneut ausführen).

Kein Zeitlimit

Gibt an, dass der Operator über kein Zeitlimit verfügt. Dieses Kontrollkästchen ist standardmäßig aktiviert.

Typ

Wählen Sie entweder "Dauer" oder "Zieldatum" aus.

Dauer/Zieldatum und -uhrzeit

- Wenn Sie "Dauer" auswählen:

  Geben Sie eine Zeitlimitdauer in Sekunden ein.

  Das richtige Format für dieses Feld ist jeder Wert, der als Ganzzahl/Long oder als Zeichenfolgenliteral angesehen werden kann (wie "10").

  Das Zeitlimit ist für jede Iteration einer Schleife spezifisch. Daher tritt eine Zeitüberschreitung nur auf, wenn eine einzelne Iteration länger dauert als die Zeitlimitdauer.

- Wenn Sie "Zieldatum" auswählen:

  Geben Sie ein Datum und Uhrzeit ein, an dem bzw. zu der der Operator das Zeitlimit erreichen soll.

  Wenn ein Zeichenfolgenliteral in dieses Feld eingegeben wird, kann dieses eines der folgenden Formate haben:

  - JVM-Format: Das Format für Datum/Uhrzeit, das die Java-Anwendung versteht. Dieses Format ist abhängig von der installierten Java-Version.
  - Wenn das JVM-Format unbekannt ist, nehmen Sie die Eingaben im Format "MM/tt/jjjj HH:mm:ss" vor.
  - Ein anderes Format muss eine CA Process Automation-Bibliotheksmethode mit dem Namen "parseDate" verwenden. Diese Methode enthält zwei Parameter: (a) das Datum als Zeichenfolgenliteral und (b) das Format, in dem die Zeichenfolge analysiert werden muss. Zum Beispiel: parseDate("10/10/2010 10:10:10"," dd/MM/yyyy HH:mm:ss") gibt das Datum: 10. Oktober 2010, 10 Uhr, 10 Minuten und 10 Sekunden zurück.

    Wenn "Ziel" als Typ ausgewählt ist, wird die Option "Zurücksetzen" (unter "Aktion") deaktiviert.

  Hinweis: Achten Sie darauf, dass diese Daten im richtigen Format eingegeben werden, da der Operator sonst diese Zeitlimit-Einstellungen ignoriert.

Aktion

Hinweise:

- Diese Option ist für Ablaufpläne nicht verfügbar.
- Wenn sich ein Operator in einer Schleife befindet und ein Zeitlimit konfiguriert wird, verhalten sich die folgenden Optionen anders. Durchgeführte Aktionen finden Sie unter Schleifen- und Zeitlimit-Szenarien (siehe Seite 31).

Wählen Sie eine der folgenden Aktionen aus:

Fortfahren

Wenn die Option ausgewählt wird, fährt der Prozess nach einem Zeitlimit in der folgenden Weise fort:

1. Der Operator bleibt nur im Ausführungsmodus.
2. Der Zeitlimit-Pfad wird verwendet.
3. Der Nachbearbeitungscode wird nur ausgeführt, wenn der Operator fertig verarbeitet ist, und nicht, wenn der Zeitlimit-Pfad verwendet wurde.

Zurücksetzen

Wenn die Option ausgewählt wird, fährt der Prozess nach einem Zeitlimit in der folgenden Weise fort:

1. Der Operator wird zurückgesetzt (das heißt, dass der Operator wieder neu ausgeführt wird).
2. Der Zeitlimit-Pfad wird verwendet.
3. Der Nachbearbeitungscode wird nur ausgeführt, wenn der Operator fertig verarbeitet wird (und nicht, wenn der Operator zurückgesetzt wurde).

Hinweis: Wenn "Zurücksetzen" ausgewählt wird, wird die Option "Zieldatum" in der Drop-down-Liste "Typ" deaktiviert.

Abbrechen

Wenn die Option ausgewählt wird, fährt der Prozess nach einem Zeitlimit in der folgenden Weise fort:

1. Der Operator wird abgebrochen.
2. Der Nachbearbeitungscode wird ausgeführt.
3. Der Zeitlimit-Pfad wird verwendet.

Verwerfen

Wenn die Option ausgewählt wird, fährt der Prozess nach einem Zeitlimit in der folgenden Weise fort:

1. Der Operator erreicht ein Zeitlimit.

2. Der Prozess wird im abgetrennten Modus weiter ausgeführt.

Hinweis: Eine Instanz eines Prozesses, der im abgetrennten Modus gestartet wurde, hat keine übergeordnete Beziehung zu dem Prozess, der ihn gestartet hat. Eine abgetrennte Prozessinstanz ist in jeder Aufrufsequenz, die aus diesem Prozess entsteht, der Stammprozess.

3. Der Nachbearbeitungscode wird sofort ausgeführt.

4. Der Zeitlimit-Pfad wird verwendet.

Schleifen- und Zeitlimit-Szenarien

Wenn sich ein Operator in einer Schleife befindet und ein Zeitlimit konfiguriert wird, treten für die ausgewählten Aktionen folgende Szenarien auf:

Aktion

Wählen Sie eine der folgenden Aktionen aus:

Fortfahren

Wenn die Option ausgewählt wird, fährt der Prozess nach einem Zeitlimit in der folgenden Weise fort:

1. Die nächste Iteration wird ausgeführt.

2. Der Nachbearbeitungscode wird nur ausgeführt, wenn die Iteration vollständig abgeschlossen ist.

Verwenden der Datensatzvariable "OverallLoopDuration", um einen Operator, der das Zeitlimit erreicht, weiterhin in einer Schleife auszuführen:

Am Ende der Ausführung enthält "OverallLoopDuration" die Anzahl von Sekunden vom Beginn der ersten Iteration bis zum Ende der Ausführung der letzten Iteration. Wenn der Operator ein Zeitlimit erreicht, enthält "OverallLoopDuration" nicht die Sekundenanzahl vom Beginn der ersten Iteration bis zu der Zeit, in der der Operator ein Zeitlimit erreicht.

Zurücksetzen

Wenn die Option ausgewählt wird, fährt der Prozess nach einem Zeitlimit in der folgenden Weise fort:

1. Die Iteration wird zurückgesetzt (das heißt, dass die bestimmte Iteration wieder neu ausgeführt wird).

2. Der Nachbearbeitungscode wird nur ausgeführt, wenn die aktuelle Iteration fertig verarbeitet wird (und nicht, wenn die Iteration zurückgesetzt wurde).

3. Die nächste Iteration wird nur ausgeführt, wenn die Iteration abgeschlossen wurde.

Hinweis: Wenn "Zurücksetzen" ausgewählt wird, wird die Option "Zieldatum" in der Drop-down-Liste "Typ" deaktiviert.

Verwenden der Datensatzvariable "OverallLoopDuration", um einen Operator zurückzusetzen, der in einer Schleife ausgeführt wird und das Zeitlimit erreicht:

Wenn Sie festlegen, dass ein Operator mit der Zeitlimitaktion "Zurücksetzen" in einer Schleife ausgeführt werden soll, überprüft CA Process Automation die Schleifenbedingung, wenn von einer Iteration in eine andere gewechselt wird. Die Schleifenbedingung wird *nicht* aktiviert, wenn eine Iteration zurückgesetzt wird. "OverallLoopDuration" enthält auch die Anzahl von Sekunden vom Beginn der ersten Iteration, einschließlich der Zeit, die in allen zurückgesetzten Iterationen aufgewendet wurde. Das Zurücksetzen von Iterationen wirkt sich nicht auf "OverallLoopDuration" aus.

Abbrechen

Wenn die Option ausgewählt wird, fährt der Ablauf nach einem Zeitlimit in der folgenden Weise fort:

1. Die Iteration wird abgebrochen.

2. Der Nachbearbeitungscode wird ausgeführt.

3. Die nächste Iteration wird ausgeführt.

Verwerfen

Wenn die Option ausgewählt wird, fährt der Prozess nach einem Zeitlimit in der folgenden Weise fort:

1. Die Iteration wird im abgetrennten Modus weiter ausgeführt.

Hinweis: Eine Instanz eines Prozesses, der im abgetrennten Modus gestartet wurde, hat keine übergeordnete Beziehung zu dem Prozess, der ihn gestartet hat, und ist in jeder Aufruffolge, die von diesem Prozess ausgeht, der Stammprozess.

2. Der Nachbearbeitungscode wird ausgeführt.

3. Die nächste Iteration wird ausgeführt.

Allgemeine Eigenschaften von Operatoren in Prozessen

Alle Operatoren haben Eigenschaften, die ihr Erscheinungsbild und Verhalten konfigurieren, wenn sie einem Prozess hinzugefügt werden.

Ein Prozess hat keine Beschränkung für die Anzahl von Operatoren, die aufgenommen werden können. Allerdings empfiehlt CA Technologies, dass ein Prozess etwa 40 bis 50 Operatoren enthalten sollte, um eine optimale Leistung zu gewährleisten. Wenn ein Prozess mehr als 40 bis 50 Operatoren enthält, sollten Sie in Erwägung ziehen, den Prozess in kleinere Komponenten aufzuteilen.

Die hier beschriebenen Eigenschaften werden nur für einen Operator in einem Prozess in den Eigenschaften "Ausführungseinstellungen", "Simulation" und "Informationen" angezeigt.

Hinweis: Operatorspezifische Eigenschaften überschreiben die auf Kategorieebene definierten Eigenschaften.

Verarbeitungsgruppe

Folgende Eigenschaften geben Bedingungen an, die erfüllt werden müssen, bevor ein Operator ausgeführt wird, und geben Aktionen an, die vor und nach dem Ausführen des Operators ausgeführt werden sollen. Sie können diese Eigenschaften in "Ausführungseinstellungen" des Operators finden.

Vorbearbeitungscode

Lässt Sie Code hinzufügen, der ausgeführt wird, bevor ein Operator ausgeführt wird. Sie können jeden beliebigen JavaScript-Code ausführen. JavaScript-Code wird ausgeführt, bevor der Operator ausgeführt wird. Vorbearbeitungscode manipuliert den Operator- und Prozessdatensatz so, dass der Datensatz als Eingabe für den Operator verwendet werden kann. Mit anderen Worten können Sie die Ausgabeparameter eines früheren Operator-Datensatzes manipulieren und sie als Eingabe für einen späteren Operator verwenden.

Vorbearbeitungscode kann verschiedene Aufgaben ausführen. Der folgende Beispielcode legt eine Variable auf Prozessebene fest:

```
if(Process.username==null)
{

Process.username="testuser";
}
```

Hinweis: Weitere Informationen zum Hinzufügen von Code finden Sie unter Operator "JavaScript ausführen" (siehe Seite 438).

Bevor der Vorbearbeitungscode abgeschlossen und der Operator ausgeführt wird, muss der Operator den Code "CanExecute = 1" erreichen. Die Prozessvariable "CanExecute" wird standardmäßig zum Operator hinzugefügt. Der Standardwert für "CanExecute" ist 1. Wenn Sie den Standard nicht ändern, wird der Operator ausgeführt. Diese Anforderung ermöglicht es Ihnen, externe Bedingungen zu überprüfen und nur dann fortzufahren, wenn die gewünschte Bedingung erfüllt ist.

Wenn Sie den CanExecute-Wert in 0 ändern (Process.CanExecute =0), wird der Operator nicht ausgeführt. CA Process Automation wartet 30 Sekunden und wiederholt dann den Vorbearbeitungscode.

Wenn kein Vorbearbeitungscode vorhanden ist, wird der Operator sofort ausgeführt. Zum Beispiel könnten Sie Vorbearbeitungscode verwenden, um Schleifenvariablen oder andere Variablen einzurichten, die als Teil des Operators verwendet werden sollen.

Für den Code, der im Operator ausgeführt wird, können Sie die folgende Syntax verwenden, um auf den Operatordatensatz zuzugreifen:

`Process[OpName].fieldname`

Der folgende Beispielcode erstellt eine Operatordatensatzvariable namens "iNow", die folgende Daten enthält:

- Den Namen des aktuellen Hosts
- Das aktuelle Datum
- Die aktuelle Zeit in einer einzelnen Zeichenfolge

```
Process[OpName].iNow = System.Host + ":" + System.Date + ":" + System.Time;
```

Nachbearbeitungscode

Lässt Sie Code hinzufügen, der ausgeführt wird, nachdem ein Operator ausgeführt wurde. Sie können Nachbearbeitungscode beispielsweise verwenden, um Schleifenvariablen zu ändern oder die Ergebnisse eines Operators zu verarbeiten.

Für den Code, der im Operator ausgeführt wird, können Sie die folgende Syntax verwenden, um auf den Operatordatensatz zuzugreifen:

`Process[OpName].fieldname`

Der folgende Beispielcode kopiert den Wert der Operatordatensatzvariable "Result" in eine Variable namens "iResult":

```
Process.iResult = Process[OpName].Result;
```

Als Aufruferanwender ausführen

Gibt an, dass der ausgewählte Operator in einem Prozess unter der Identität des autorisierten Anwenders, der ihn gestartet hat, ausgeführt werden muss. Diese Anforderung trifft unabhängig davon zu, ob der gesamte Prozess als Verantwortlicher ausgeführt wird oder nicht. "Als Aufruferanwender ausführen" lässt Prozessdesigner Prozesse ausführen, die:

- Ein eigenständiges Automatisierungsobjekt liefern (Ausführung als der Verantwortliche)
- Die Steuerung von Zugriffsrechten für Teilen des Prozesses (wie untergeordnete Prozesse und Kontaktpunkte) erfordern

Simulationseigenschaften

Die Simulationseigenschaften lassen Sie konfigurieren, wie die Ausführung eines Operators in einem Prozess simuliert werden soll. Die Simulation kann verwendet werden, um Verzweigungen eines Prozesses zu testen oder es zu ermöglichen, dass ein Operator im Rahmen der normalen Ausführung übersprungen wird, ohne dass der Prozess umgeleitet werden muss.

Sie konfigurieren Simulationseigenschaften für einen neuen Prozess. Für jeden Operator in einem Prozess können Sie angeben, dass die auf Prozessebene konfigurierte Moduseinstellung übernommen werden soll, oder eine andere Moduseinstellung angeben. Doppelklicken Sie auf einen Operator, um den Bereich "Operatoreigenschaften" anzuzeigen. Blenden Sie dann die Registerkarte "Simulation" ein, um die zu konfigurierenden Einstellungen anzuzeigen.

Modus

Simulationsmodi sind verfügbar, wenn Sie das Kontrollkästchen "Simulationsoptionen im Prozess überschreiben" aktivieren.

Vom Prozess übernehmen

Gibt an, dass die Einstellung, die für den übergeordneten Prozess des Operators konfiguriert ist (Deaktiviert, Lokal oder Entfernt), verwendet werden soll.

Deaktiviert

Deaktiviert die Simulation und aktiviert die normale Verarbeitung des Operators. Die Endbedingung ist auf "Abgeschlossen" festgelegt, "Verzögerung" ist auf 0 festgelegt, und "Vor- und Nachbearbeitungscodes evaluieren" ist deaktiviert.

Lokal

Deaktiviert den Operator, sodass dieser nicht verarbeitet wird. CA Process Automation ruft den zugeordneten Operator nicht auf oder überprüft die Operatorparameter nicht. Zu den Parameterprüfungen gehört die Überprüfung auf ein Anwendungsprogramm oder die Validierung des Ausführungskontaktpunkts eines Operators.

Entfernt

Bewirkt, dass der zugeordnete Operator aufgerufen wird. Der Operator überprüft die Parameter, bevor er das Ergebnis zurückgibt, führt die Operation aber nicht tatsächlich aus. Wenn die Parameter nicht korrekt sind, schlägt der simulierte Operator unabhängig vom angegebenen Ergebnis fehl. Wenn die Parameter richtig sind, gibt der Operator das angegebene Ergebnis zurück.

Verzögerung

Gibt die Anzahl von Sekunden an, um die ein Prozess verzögert werden soll, um den Zeitraum zu simulieren, den die normale Verarbeitung des Operators in Anspruch nimmt.

Endbedingung

Gibt die Abbruchbedingung für den simulierten Operator an. Sie können diese Option verwenden, um in einem Prozess unterschiedliche Verzweigungen zu testen oder Fehler darin zu beheben. Die eigentlichen Bedingungen hängen vom Operator ab.

- Die folgenden Endbedingungen gelten für den Modus "Deaktiviert": Abgeschlossen.
- Die folgenden Endbedingungen gelten für den Modus "Lokal": Abgeschlossen, Fehlgeschlagen, Zeitlimit.
- Die folgenden Bedingungen gelten für den Modus "Entfernt": Abgeschlossen, Fehlgeschlagen, Zeitlimit, Anwenderspezifisches Ergebnis.

Abgeschlossen

Bewirkt, dass die standardmäßige Ausgangsverknüpfung für eine erfolgreiche Ausführung für den Operator verarbeitet wird. Die Ergebnisvariable im Operatordatensatz wird auf 1 festgelegt. Ein positiver Ganzzahl-Wert aktiviert die standardmäßige Ausgangsverknüpfung für eine erfolgreiche Ausführung.

Fehlgeschlagen

Bewirkt, dass die standardmäßige Ausgangsverknüpfung für eine fehlgeschlagene Ausführung für den Operator verarbeitet wird. Die Ergebnisvariable im Operatordatensatz wird auf 0 festgelegt. Null oder ein negativer Ganzzahl-Wert aktiviert die standardmäßige Ausgangsverknüpfung für eine fehlgeschlagene Ausführung.

Zeitlimit

Führt dem Operator dazu, den Zeitlimit-Pfad zu verwenden, wenn die angegebene Zeit vergangen ist. Die Ergebnisvariable im Operatordatensatz wird als Zeitüberschreitung angezeigt.

Anwenderspezifisches Ergebnis

Gibt den von der Ergebnisvariablen im Operatordatensatz zurückgegebenen ganzzahligen Wert an. Sie können diesen Parameter auf einen beliebigen Wert (positiv oder negativ) setzen, um einen anwenderspezifischen Port zu aktivieren, die auf einen bestimmten Wert testet.

Diese Option ist nur verfügbar, wenn "Anwenderspezifisches Ergebnis" als Endbedingung ausgewählt ist.

Vor- und Nachbearbeitungscodes evaluieren

Zeigt an, ob Vor- und Nachbearbeitungscode während der Operatorsimulation ausgewertet werden sollen.

Ausgewählt

Gibt an, dass der Code ausgewertet werden soll, und verhindert die möglichen Auswirkungen eines ignorierten Codes.

Gelöscht

Gibt an, dass der Code nicht ausgewertet werden soll.

Informationseigenschaften

Die Symboleigenschaften bestimmen den Namen des Operators sowie die visuelle Darstellung des Operators und der zugehörigen Kommentare. Standardeinstellungen (für den gesamten Prozess) werden über die Schaltfläche "Eigenschaften" in der Symbolleiste "Designer" festgelegt.

Objektvoreinstellungen überschreiben

Aktivieren Sie dieses Kontrollkästchen, um die standardmäßigen Prozesseinstellungen zu überschreiben und spezifische Einstellungen für diesen Operator zu konfigurieren.

Symbol

Name

Gibt den Namen des Operators an. Die Eigenschaft "Name" ist besonders wichtig, wenn ein Operator konfiguriert wird, auf den andere Operatoren verweisen werden. Die Eigenschaft "Name" wird mit der folgenden Syntax in einem Ausdruck verwendet, um in einem Prozess auf Variablen im Operatordatensatz eines anderen Operators zuzugreifen:

```
Process.Operator_name.variable_name
```

Operatornamen müssen innerhalb eines Prozesses eindeutig sein.

Standardsymbol verwenden

Wenn diese Option aktiviert ist, verwendet der Operator das Standardsymbol. Deaktivieren Sie dieses Kontrollkästchen, um statt des Standardsymbols ein anwenderspezifisches Symbol zu verwenden.

Durchsuchen

Klicken Sie auf diese Option, um das anwenderspezifische Symbol auszuwählen, das Sie für diese Instanz des Operators verwenden möchten.

Beschriftungsanzeige

Beschriftungen anzeigen

Zeigt Symbolinformationen für den Operator an, den Sie in das Feld "Name" eingeben. Wählen Sie eine der folgenden Optionen:

Gekürzt

Zeigt nur einen Teil des Feldes "Name" neben dem Operatorsymbol an.

Long

Zeigt die gesamte Länge des Feldes "Name" neben dem Operatorsymbol an.

Aus

Es wird keine Beschriftung angezeigt.

Beschriftungsquelle

Objektname

Wählen Sie diese Option aus, um den Namen des Operators als Beschriftung anzuzeigen.

Kommentare

Wählen Sie diese Option aus, um die Operator-Kommentare (die Sie in das Textfeld "Kommentare" eingeben) als Beschriftung anzuzeigen.

Beschriftungsfarben

Klicken Sie auf "Farbe auswählen", um eine Farbe für die folgenden Teile der Beschriftung auszuwählen:

- Textfarbe
- Hintergrundfarbe
- Rahmenfarbe

Schriftarten

Konfigurieren Sie die Schrifteigenschaften der Symbolbeschriftung für den Operator: Schriftfamilie, Schriftstil und Schriftgröße.

Hinweis: Wählen Sie Schriftarten aus, die auf Computern, die CA Process Automation hosten, in der Regel verfügbar sind.

Vorschau

Überprüfen Sie Ihre Schriftauswahl der Symbolbeschriftung für den Operator, bevor Sie sie anwenden.

Kommentare

Geben Sie Kommentare ein, die Sie für den Operator anzeigen möchten. Wenn "Beschriftungsquelle" den Wert "Kommentare" enthält, wird dieser Text in der Beschriftung neben dem Operator angezeigt.

Allgemeine Eigenschaften von Operatoren in Ablaufplänen

Operatoren in einem Ablaufplan werden entsprechend den angegebenen Kalender- und Uhrzeitbedingungen gestartet. Eigenschaftsgruppen für jeden dem Ablaufplan hinzugefügten Operator verknüpfen diesen mit Kalenderregeln und anderen Uhrzeitbedingungen. Klicken Sie auf das Eigenschaftssymbol im Ablaufplan-Editor, um die Eigenschaften eines Operators anzuzeigen. Allgemeine Ablaufplaneigenschaften werden auf der Registerkarte "Allgemein" angezeigt. Operatorspezifische werden auf der Registerkarte "Spezifisch" angezeigt.

Geben Sie die Zeit zum Starten des Operators im ersten Drop-down-Menü in der Drop-down-Liste "Startzeit" ein. Geben Sie Wiederholungsintervalle über das Kontrollkästchen "Wiederholungsintervall (Minuten)" an. Aktivieren Sie dieses Kontrollkästchen, um den Operator in festen Intervallen auszuführen. Wählen Sie im Drop-down "Endzeit" eine Zeit aus, an dem die wiederholte Ausführung des Operators gestoppt werden soll.

Hinweis: Weitere Informationen über Ablaufpläne finden Sie im *Handbuch für Inhaltsdesign*.

Kalender-Einstellungen

Die Eigenschaften für die Kalender-Einstellungen ermöglichen Ihnen, Daten aus einem vordefinierten Kalender in einem Ablauf einzuschließen.

Einschließungskalender

Klicken Sie auf das Kalendersymbol, um einen vordefinierten in Ihren Ablaufplan einzuschließenden Kalender auszuwählen.

Ausschlusskalender

Klicken Sie auf das Kalendersymbol, um einen vordefinierten in Ihrem Ablaufplan auszuschließenden Kalender auszuwählen.

Tage pro Aufschub

Die Anzahl von Tagen, um die ein geplantes Datum verschoben wird, wenn das geplante Datum auf ein geschlossenes Datum fällt. Der Aufschub kann negativ oder 0 sein. Wenn dieser Wert negativ ist, wird das Datum nach vorne verschoben. Wenn dieser Wert gleich 0 ist, werden geschlossene Daten einfach übersprungen, ohne dass die Aufgabe erneut geplant wird.

Keine ausgeschlossenen Tage

Aktivieren Sie dieses Kontrollkästchen, damit nur offene Tage gezählt werden, wenn ein geplantes Datum verschoben wird, um ein geschlossenes Datum zu vermeiden.

Maximale Aufschübe

Wenn das ursprüngliche geplante Datum auf einen geschlossenen Tag fällt und die Aufgabe erneut geplant wird, fällt das neue Datum möglicherweise ebenfalls auf ein geschlossenes Datum. Dieser Parameter gibt der größtmögliche Anzahl von Aufschüben an, die zulässig sind.

Nur manuelle Auswahl

Wenn ein Kalender in einem Ablaufplan nicht angegeben ist, nimmt CA Process Automation an, dass das Element für jeden Tag zu planen ist. Dies gilt nicht, wenn diese Option ausgewählt ist. Wenn diese Option ausgewählt ist, planen Sie Datumswerte für die Ausführung (auf der Registerkarte "Manuell eingeschlossene Daten").

Manuell eingeschlossene Daten

Die Eigenschaften "Manuell eingeschlossene Daten" ermöglichen es Ihnen, manuell Datumsangaben zu einem Ablaufplan hinzuzufügen. Diese Eigenschaften zeigen eine Liste der Daten an, die manuell geplant werden. Sie zeigen auch eine Liste der Daten an, die von den Kalenderregeln angegeben werden. Hier hinzugefügte Daten überschreiben die geschlossenen Tage, die von den Eigenschaften "Manuell ausgeschlossene Daten" angegeben werden.

Wenn ein Ablaufplan keinen Kalender angibt, werden die Eigenschaften "Manuell eingeschlossene Daten" verwendet, um Daten zu planen, wenn das Kontrollkästchen "Nur manuelle Auswahl" in den Eigenschaften "Kalendereinstellungen" ausgewählt ist.

Klicken Sie auf die entsprechende Optionsschaltflächen, um Listenelemente hinzuzufügen, zu entfernen oder neu anzuordnen.

Element hinzufügen

Fügt der Liste ein Datum hinzu. Um das Datum festzulegen, klicken Sie auf den Eintrag. Klicken Sie dann auf das Kalendersymbol, um die einzuschließenden Daten auszuwählen.

Element löschen

Entfernt ein ausgewähltes Datum aus der Liste.

Nach oben verschieben

Verschiebt ein ausgewähltes Datum in der Liste nach oben.

Nach unten verschieben

Verschiebt ein ausgewähltes Datum in der Liste nach unten.

Manuell ausgeschlossene Daten

Mithilfe der Eigenschaften "Manuell ausgeschlossene Daten" können Sie manuell Datumswerte aus einem Ablaufplan entfernen. Die Eigenschaft "Manuell ausgeschlossene Daten" listet Daten auf, die unter keinen Umständen, ungeachtet aller anderen Regeln oder Bedingungen, geplant werden dürfen.

Klicken Sie auf die entsprechende Optionsschaltfläche, um Listenelemente hinzuzufügen, zu entfernen oder neu anzuordnen.

Element hinzufügen

Fügt der Liste auszuschließende Daten hinzu. Um das Datum festzulegen, klicken Sie auf den Eintrag. Klicken Sie dann auf das Kalendersymbol, um die einzuschließenden Daten auszuwählen.

Element löschen

Entfernt ein ausgewähltes Datum aus der Liste.

Nach oben verschieben

Verschiebt ein ausgewähltes Datum in der Liste nach oben.

Nach unten verschieben

Verschiebt ein ausgewähltes Datum in der Liste nach unten.

Name der Aufgabe

Der Name der Aufgabe gibt den Namen der anwenderdefinierten Aufgabe an. Wenn Sie einem Ablaufplan einen Operator oder Prozess hinzufügen, können Sie in diesem Feld einen anwenderspezifischen Namen angeben. Der anwenderspezifische Aufgabenname gilt für entsprechende Laufzeitaufgabeninstanzen.

Gewöhnliche Ausgabeparameter für alle Operatoren

Alle Operatoren enthalten folgende Ausgabeeigenschaften. Weitere Ausgabeparameter werden für jeden Operator angegeben.

StartTime

Die Zeit, zu der der Operator im Prozess oder Ablaufplan startete.

StartDate

Das Datum, zu dem der Operator im Prozess oder Ablaufplan startete.

Ursache

Gibt den Grund an, wenn der Operator nach der Ausführung fehlschlägt.

Ergebnis

Gibt das Ergebnis der Operator-Ausführung an.

Kapitel 2: Standardoperatoren

Verwenden Sie die Standardoperatoren, um die Workflows in Prozessen zu steuern.

Operator "Start"

Verwenden Sie den "Start"-Operator, um den Workflow in einem Prozess zu starten. Standardmäßig ist der "Start"-Operator automatisch in einem Prozess enthalten.

Sie können mehr als einen "Start"-Operator einem Prozess hinzufügen. Jeder "Start"-Operator in einem Prozess startet seinen eigenen Workflow, wenn ein Koordinationsrechner den Prozess startet.

Sie können auch einen Operator "Start" hinzufügen, um eine zyklische Sequenz von Operatoren zu beenden. Ein "Start"-Operator, der einen Workflow beendet, reinitialisiert die Operatoren im Workflow. Der Operator führt die Verarbeitung zurück zum anfänglichen Startsymbol für die Sequenz von Operatoren, die zwischen den "Start"-Operatoren ausgeführt wurden.

Eingabeparameter

Doppelklicken Sie auf den Operator "Start", um den Namen und die Darstellung mithilfe der Informationseigenschaften (siehe Seite 38) zu konfigurieren.

Operator "Kommentar"

Der Operator "Kommentar" fügt dem Prozess Kommentare hinzu. Kommentare sind wichtig für das Dokumentieren von Schritten in einem Prozess und bieten mehr Platz als Beschriftungen.

Gehen Sie folgendermaßen vor::

1. Ziehen Sie den Operator "Kommentar" aus dem Standardordner auf einen Bereich der Editor-Registerkarten ("Haupteditor", "Ausnahmen-Handler", "Prozessspurwechsel-Handler").
2. Doppelklicken Sie den Kommentartext, um die Kommentareigenschaften anzuzeigen.

Eingabeparameter

Hintergrund

Rahmenfarbe

Klicken Sie auf die Drop-down-Liste "Farbe auswählen", um die Farbe für den Kommentarrahmen auszuwählen.

Hintergrundfarbe

Wählen Sie die Farbe und Transparenz des Hintergrunds für den Kommentartext aus.

- Transparent für einen transparenten Hintergrund.
- Undurchsichtig für einen farbigen Hintergrund. Wenn Sie diese Option auswählen, können Sie auch die für den Hintergrund verwendete Farbe ändern.

Kommentare

Ermöglicht es Ihnen, den Kommentartext einzugeben oder zu ändern.

- Konfigurieren Sie die Schrifteigenschaften: Schriftart, Schriftstil, Größe, Effekte (fett, kursiv und unterstrichen), Farbe und Hervorhebung.

  Wählen Sie Schriftarten, die höchstwahrscheinlich auf den Computern installiert sind, die CA Process Automation hosten.
- Konfigurieren Sie die Ausrichtung: linke, rechte oder zentrierte horizontale Ausrichtung des Kommentartextes. Aufzählungszeichen und nummerierte Listen sind ebenfalls verfügbar.
- Klicken Sie auf "Hyperlink", um den ausgewählten Text in einen Hyperlink umzuwandeln.
- Klicken Sie auf "Quellcodebearbeitung", um in den Bearbeitungsmodus der Quelle zu wechseln.

Operator "Bei Erfolg beenden"

Der Operator "Bei Erfolg beenden" beendet einen Prozess und bestimmt diesen als Erfolg. Ein Operator "Bei Erfolg beenden" kann einen Prozess beenden:

- Am Ende einer Sequenz von Operatoren auf der Registerkarte "Haupteditor".
- In einer Ausnahme auf der Registerkarte "Ausnahmen-Handler".
- In einem Prozessspurwechsel auf der Registerkarte "Prozessspurwechsel-Handler".

Ein Operator "Bei Erfolg beenden" kann entweder als Operator "Bei Erfolg beenden" oder Operator "Bei Fehlschlag beenden" konfiguriert werden (über seine Eigenschaften). Wenn ein Prozess ausgeführt wird, legt der Operator "Bei Erfolg beenden" die Ergebnisvariable für einen Operatordatensatz standardmäßig auf 1 fest. Sie können den positiven Standardwert durch einen negativen überschreiben, um den Operator "Bei Erfolg beenden" in einen Fehler zu ändern.

Eingabeparameter

Ergebnis

Gibt einen Ergebnisparameter an. Der Ergebnisparameter ist ein Ganzzahl-Ausdruck, der verwendet wird, um zu entscheiden, ob der Ablauf richtig (positiver Wert) oder fehlerhaft (Null oder negativer Wert) endete.

Wenn Sie einen Endtyp "Bei Erfolg beenden" auswählen, lautet der Standardwert 1. Wenn Sie "Bei Fehlschlag beenden" auswählen, lautet der Standardwert -1. Sie können auch einige andere Ganzzahl-Werte eingeben oder einen Ausdruck eingeben, der einen berechneten Wert für den Ergebniscode zur Laufzeit zurückgibt.

Der Ergebniswert für den Operator "Bei Erfolg beenden", der eine Instanz eines Prozesses beendet, wird im Prozessdatensatz der Instanz in der Ergebnisvariable gespeichert.

End Type

Wählen Sie eine der folgenden Optionen:

Bei Erfolg beenden

Bearbeitet eine normale Beendigung für einen Workflow. Dies legt den Code "Ergebnis" auf 1 fest. Verwenden Sie eine positive Ganzzahl, wenn Sie den Ergebniswert ändern, damit der Wert mit dem einer normalen Beendigung konsistent ist.

Bei Fehlschlag beenden

Verarbeitet ein unerwartetes Ende für einen Ablauf. Diese Option legt den Wert "Ergebnis" auf -1 fest. Geben Sie eine negative Ganzzahl ein, wenn Sie den Wert "Ergebnis" ändern, um einem unerwarteten Ende zu entsprechen.

Aufrufschleife abbrechen

Wenn der Ablauf durch einen anderen Prozess aufgerufen wird, wird durch Aktivieren dieses Kontrollkästchens eine Aufrufschleife unterbrochen. Durch Deaktivieren dieses Kontrollkästchens kann eine Aufrufschleife fortgesetzt werden. Dieses Kontrollkästchen gilt nur, wenn der Ablauf aus einer Schleife in einem *anderen* Prozess aufgerufen wurde.

Derzeit ausgeführte Aufgaben ignorieren (sofort anhalten)

Beendet einen Ablauf sofort, ohne darauf zu warten, dass andere Operatoren mit der Bearbeitung aufhören. Deaktivieren Sie dieses Kontrollkästchen, um darauf zu warten, dass die Bearbeitung durch Operatoren beendet wird, bevor Sie den Ablauf beenden.

Operator "Bei Fehlschlag beenden"

Der Operator "Bei Fehlschlag beenden" beendet einen Prozess und bestimmt ihn als Fehler. Ein Operator "Bei Fehlschlag beenden" kann einen Prozess beenden:

- Am Ende einer Sequenz von Operatoren auf der Registerkarte "Haupteditor".
- In einer Ausnahme auf der Registerkarte "Ausnahmen-Handler".
- In einem Prozessspurwechsel auf der Registerkarte "Prozessspurwechsel-Handler".

Ein Operator "Bei Fehlschlag beenden" kann entweder als Operator "Bei Erfolg beenden" oder Operator "Bei Fehlschlag beenden" konfiguriert werden (über seine Eigenschaften). Wenn ein Prozess ausgeführt wird, legt der Operator "Bei Fehlschlag beenden" die Ergebnisvariable für einen Operatordatensatz standardmäßig auf -1 fest. Sie können den negativen Standardwert durch einen positiven überschreiben, um den Operator "Bei Fehlschlag beenden" in einen Erfolg zu ändern.

Eingabeparameter

Ergebnis

Gibt einen Ergebnisparameter an. Der Ergebnisparameter ist ein Ganzzahl-Ausdruck, der verwendet wird, um zu entscheiden, ob der Ablauf richtig (positiver Wert) oder fehlerhaft (Null oder negativer Wert) endete.

Wenn Sie einen Endtyp "Bei Fehlschlag beenden" auswählen, lautet der Standardwert -1. Wenn Sie "Bei Erfolg beenden" auswählen, lautet der Standardwert 1. Sie können auch einige andere Ganzzahl-Werte eingeben oder einen Ausdruck eingeben, der einen berechneten Wert für den Ergebniscode zur Laufzeit zurückgibt.

Der Ergebniswert für den Operator "Bei Fehlschlag beenden", der eine Instanz eines Prozesses beendet, wird im Prozessdatensatz der Instanz in der Variable "Ergebnis" gespeichert.

End Type

Wählen Sie eine der folgenden Optionen:

Bei Erfolg beenden

Bearbeitet eine normale Beendigung für einen Workflow. Dies legt den Code "Ergebnis" auf 1 fest. Verwenden Sie eine positive Ganzzahl, wenn Sie den Ergebniswert ändern, damit der Wert mit dem einer normalen Beendigung konsistent ist.

Bei Fehlschlag beenden

Verarbeitet ein unerwartetes Ende für einen Ablauf. Diese Option legt den Wert "Ergebnis" auf -1 fest. Geben Sie eine negative Ganzzahl ein, wenn Sie den Wert "Ergebnis" ändern, um einem unerwarteten Ende zu entsprechen.

Aufrufschleife abbrechen

Wenn der Ablauf durch einen anderen Prozess aufgerufen wird, wird durch Aktivieren dieses Kontrollkästchens eine Aufrufschleife unterbrochen. Durch Deaktivieren dieses Kontrollkästchens kann eine Aufrufschleife fortgesetzt werden. Dieses Kontrollkästchen gilt nur, wenn der Ablauf aus einer Schleife in einem *anderen* Prozess aufgerufen wurde.

Derzeit ausgeführte Aufgaben ignorieren (sofort anhalten)

Beendet einen Ablauf sofort, ohne darauf zu warten, dass andere Operatoren mit der Bearbeitung aufhören. Deaktivieren Sie dieses Kontrollkästchen, um darauf zu warten, dass die Bearbeitung durch Operatoren beendet wird, bevor Sie den Ablauf beenden.

Operator "Prozessfortschritt"

Der Operator "Prozessfortschritt" ermöglicht es Ihnen, den Fortschritt in verschiedenen Phasen eines Prozesses festzulegen. Berechtigte Anwender können den Fortschritt des Prozesses folgendermaßen überwachen:

- Prozessdatensatz
- Anwenderspezifische Berichte
- Die Tabelle "Prozessinstanzen" auf der Registerkarte "Vorgänge"

Hinweis: Weitere Informationen über berechtigte Anwender finden Sie im Abschnitt *Berechtigungen nach Registerkarten* im *Handbuch für Inhaltsadministratoren*.

Eingabeparameter

Prozessfortschritt

Gibt den Abschlussfortschritt eines Prozesses als Prozentsatz im Bereich von 0 bis 100 an. Wenn die Ausführung eines Prozesses den Operator "Bei Erfolg beenden" erreicht, dann hat der Prozessfortschritt den Wert 100. Wenn ein Prozess fehlschlägt, nachdem er einen Fortschritt von 50 erreicht hat, zeigt der Prozessdatensatz den Wert 50 an.

Ausgabe

Fortschritt

Zeigt den Prozentsatzabschluss eines Prozesses unter "Fortschritt" (Variablenname) im Prozessdatensatz an.

Operator "Und"

Der Operator "Und" definiert einen Synchronisationspunkt zwischen allen für ihn angegebenen Eingangsverknüpfungen. Ausgangsverknüpfungen von einem Operator "Und" werden erst aktiviert, nachdem alle Eingangsverknüpfungen aktiviert worden sind. Verwenden Sie den Operator "Und", um mehrere Verzweigungen eines Prozesses mit einer logischen Und-Bedingung zu synchronisieren, wenn zwei oder mehr getrennte Verzweigungen eines Ablaufs abgeschlossen sein müssen, bevor mit der Ausführung einer oder mehrerer zusätzlicher Verzweigungen begonnen werden kann.

Sie können einen Operator "Und" in einem Prozess auf verschiedene Arten einschließen.

Gehen Sie folgendermaßen vor::

1. Ziehen Sie den Operator "Und" aus dem Standardordner in eine der Editor-Registerkarten ("Haupteditor", "Ausnahmen-Handler", "Prozessspurwechsel-Handler").
2. Verknüpfen Sie mindestens einen Eingabeoperator, der von Operator "Und" synchronisiert werden kann.
3. Verknüpfen Sie einen oder mehrere Ausgabe-Operatoren, die nach dem Abschluss des Operators "Und" ausgeführt werden sollen.

Parameter

Der Operator "Und" enthält keine Parameter.

Operator "Oder"

Der Operator "Oder" definiert einen Synchronisationspunkt zwischen allen angegebenen Eingangsverknüpfungen. Ausgangsverknüpfungen eines Operators "Oder" werden aktiviert, nachdem mindestens eine von einer beliebigen Anzahl von Eingangsverknüpfungen für den Operator aktiviert wurde. Mindestens eine von zwei oder mehr getrennten Verzweigungen eines Ablaufs, der zu einem Operator "Oder" führt, müssen abgeschlossen werden, bevor mit der Ausführung einer oder mehrerer Ausgangsverzweigungen begonnen wird.

Der Operator "Oder" kann zu Sequenzen von Operatoren in einem Prozess auf einer der Editor-Registerkarten hinzugefügt werden ("Haupteditor", "Ausnahmen-Handler", "Prozessspurwechsel-Handler"). Fügen Sie einen Operator "Oder" in einen Prozess ein, indem Sie ihn aus dem Standardordner in einen dieser drei Editoren ziehen. Verknüpfen Sie einen oder mehrere Eingabe-Operatoren, die durch den Operator "Oder" synchronisiert werden sollen, und verknüpfen Sie einen oder mehrere Ausgabe-Operatoren, die nach dem Abschluss des Operators "Oder" ausgeführt werden sollen.

Parameter

Der Operator "Oder" enthält keine Parameter.

Operator "Zurücksetzen"

Verwenden Sie den Operator "Zurücksetzen", um ausgewählte Operatoren (normalerweise einen Operator, der eine Ausnahme verursacht hat) in einem unterbrochenen Prozess so auf ihren Anfangsstatus zurückzusetzen. Diese Zurücksetzen-Operatoren agieren, als ob sie nicht ausgeführt worden wären, und sie setzen dann den Prozess fort.

Dieser Operator ermöglicht es Anwendern außerdem, eine Ausnahme zu ignorieren und einen Prozess auf jeden Fall fortzusetzen. Der Operator "Zurücksetzen" ermöglicht dem Anwender, einen Operator im Simulationsmodus festzulegen und dann im Prozess mit dem simulierten Operator fortzufahren.

Eingabeparameter

Operatoren-Liste

Klicken Sie auf "Hinzufügen", um einen zurückzusetzenden Operator hinzuzufügen. Über ein Dropdown-Menü können Sie einen der verfügbaren Operatornamen im aktuellen Prozess auswählen. Mehrere Operatoren können hinzugefügt werden, die dann hinzugefügt, gelöscht oder sortiert werden können.

Sie können auch einen Ausdruck eingeben (anstatt einen Operator aus dem Drop-down-Menü auszuwählen), der während des Betriebs zu einer Zeichenfolge (ein Operatorname) oder einem Werte-Array (einer Liste mit Operatornamen) aufgelöst wird.

Mit Ergebnis fortfahren

Ausgewählt

Das Drop-down-Menü "Endbedingung" und das Kontrollkästchen "Vor- und Nachbearbeitungscodes evaluieren" sind verfügbar.

Gelöscht

Wenn zur Laufzeit eine Fehlerbedingung auftritt, setzt CA Process Automation die ausgewählten Operatoren zurück. CA Process Automation setzt dann den Prozessablauf fort

Endbedingung

Wenn Sie das Kontrollkästchen "Mit Ergebnis fortfahren" aktivieren, wird das Dropdown-Menü "Beendigungsbedingung" verfügbar und enthält die folgenden Optionen:

Erfolgreich

CA Process Automation nimmt an, dass die ausgewählten Operatoren erfolgreich sind, wenn keine Fehlerbedingung zur Laufzeit erfüllt wird. CA Process Automation setzt dann den Rest des Prozessablaufs fort.

Nicht erfolgreich

CA Process Automation nimmt an, dass die ausgewählten Operatoren fehlgeschlagen sind, wenn eine Fehlerbedingung zur Laufzeit erfüllt wird. CA Process Automation setzt dann den Rest des Prozessablaufs fort.

Vor- und Nachbearbeitungscodes evaluieren

Aktivieren Sie dieses Kontrollkästchen, um Vor- und Nachbearbeitungscodes zu evaluieren.

So funktioniert der Operator "Zurücksetzen" mit dem Operator "Schleife"

Der Operator "Zurücksetzen" funktioniert folgendermaßen mit dem Operator "Schleife".

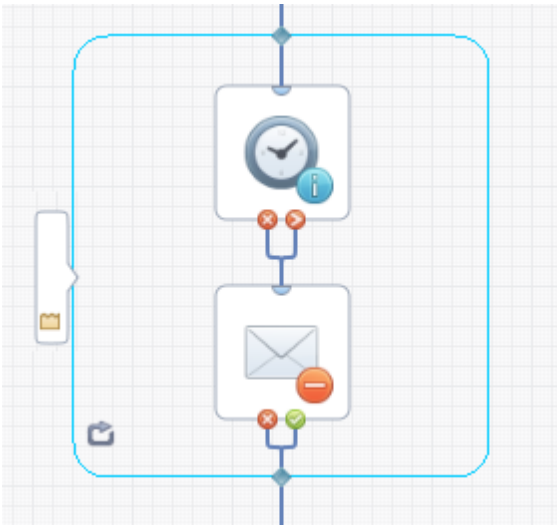
1. Der Operator "Zurücksetzen" ermöglicht das Zurücksetzen des Operators "Schleife". Der Operator "Zurücksetzen" setzt den Operator "Schleife" folgendermaßen zurück:
   - Setzt alle Operatoren im Operator "Schleife" zurück.
   - Setzt den Operator "Schleife" zur ersten Iteration zurück.
2. Nach dem Zurücksetzen startet der Operator "Schleife" von der ersten Iteration neu.

Hinweis: Der Operator "Schleife" unterstützt keine Simulation. Der Operator "Zurücksetzen" setzt den Operator "Schleife" immer zurück, unabhängig von den Werten der Felder:

- Mit Ergebnis fortfahren
- Endbedingung
- Vor- und Nachbearbeitungscodes evaluieren

Schleifenoperator

Der Operator "Schleife" stellt eine geschlossene Sequenz von Operatoren in einem Prozess in eine Schleife, die für eine festgelegte Anzahl oder unendlich ausgeführt wird. Sie können ihn in einem Prozess platzieren und die Größe des Feldes ändern, um eine beliebige Anzahl an Operatoren in der Sequenz unterzubringen.



Der Operator "Schleife" kann eine Sequenz von Operatoren in einen Prozess im Bereich "Haupteditor", im Bereich "Ausnahmen-Handler" oder im Bereich "Prozessspurwechsel-Handler" des Designers einschließen.

So fügen Sie einen Operator "Schleife" in einen Prozess ein

1. Ziehen Sie den Operator "Schleife" aus dem Standardordner zu einem Editor.
2. Ziehen Sie eine oder mehrere Eingabeverknüpfungen in das Eingabeportal und einen oder mehrere Ausgabeverknüpfungen aus dem Ausgabeportal.
3. Fügen Sie dem Feld "Schleife" Schleifen-Operatoren hinzu.
4. Verknüpfen Sie das Eingabeportal mit dem ersten Operator in der Schleifensequenz.
5. Verknüpfen Sie den letzten Operator in der Sequenz mit dem Ausgabeportal.

Eingabeparameter

Wiederholungsanzahl

Gibt an, wie oft ein Operator wiederholt wird. Folgende Optionen stehen zur Verfügung:

- Für diesen Wert kann eine Ganzzahl oder ein CA Process Automation-Ausdruck angegeben werden, der zur Laufzeit eine Ganzzahl zurückgibt. Der Standardwert 1 führt eine Schleife in einem Workflow einmal für einen Operator aus. Um eine Endlosschleife auszuführen, klicken Sie auf das Kontrollkästchen "Endlosschleife".
- Es kann auch ein boolescher Ausdruck verwendet werden. Solange der Ausdruck als "Wahr" bewertet wird, führt ein Operator in einem Workflow eine dauerhafte Schleife aus. Wenn der Ausdruck "Falsch" ist, wird der Operator beendet.

Dieser Wert kann auch mithilfe der Schleifenvariablen im Datensatz des Operators "Schleife" angegeben werden.

- CurrentLoopIteration: Ein Schleifenzähler, der während der ersten Iteration bei 0 beginnt und für jede zusätzliche Iteration um 1 erhöht wird. Diese Variable am Anfang und am Ende jeder Iteration aktualisiert.

  Wenn der Operator so konfiguriert ist, dass er dreimal in einer Schleife ausgeführt wird, ist "CurrentLoopIteration" am Ende der Ausführung aller Iterationen gleich:

  - 0 in iteration 1

  - 1 in Iteration 2

- 2 in Iteration 3

- 3 in der letzten Iteration, die nicht ausgeführt wird, da sie die Schleifenbedingung verletzt.

– OverallLoopDuration: Ein Schleifenzähler, der die Zeit (in Sekunden) angibt, die seit dem Start der ersten Iteration der Schleife vergangen ist. Diese Variable wird am Anfang und Ende jeder Iteration aktualisiert und schließt Verzögerungen ein, die zwischen Iterationen der Schleife festgelegt sind.

Legen Sie die Wiederholungsanzahl auf folgende Werte fest:

Process[OpName].CurrentLoopIteration < *x*

wobei

x angibt, wie oft der Operator ausgeführt werden soll.

Oder legen Sie die Wiederholungsanzahl auf folgende Werte fest:

Process[OpName].OverallLoopDuration < *x*

wobei

x die Anzahl von Sekunden angibt, die der Operator in einer Schleife ausgeführt werden soll. Der Operator wird nicht bei der angegebenen Anzahl von Sekunden beendet, wenn er sich mitten in einer Iteration befindet. Wenn "OverallLoopDuration" größer ist als die angegebene Anzahl von Sekunden, führt der Operator stattdessen die Iteration nicht weiter aus.

CA Process Automation prüft die Schleifenbedingung zwischen Iterationen.

Endlosschleife

Wenn diese Option ausgewählt ist, wird "Wiederholungsanzahl" ignoriert und eine Endlosschleife erzeugt. Der Operator wird so lange wiederholt, bis eine der beiden Bedingungen eintritt:

- Der Prozess wird unterbrochen.
- Die Schleife wird von einer anderen Verzweigung des Prozesses angehalten (indem eine Befehlsverknüpfung zur Beendigung einer Schleife für den Operator "Schleife" ausgeführt wird).

Verzögerung zwischen Iterationen

Gibt die Verzögerung in Sekunden zwischen jeder Schleifenwiederholung an.

While-Schleife

Wenn diese Option ausgewählt ist, verhält sich der Operator "Schleife" wie eine *While*-Schleife. Wenn diese Option nicht ausgewählt ist, verhält sich der Operator "Schleife" wie eine *Do-While*-Schleife.

While-Schleife

Der Operator "Schleife" überprüft die Schleifenbedingung, die im Feld "Wiederholungsanzahl" angegeben ist, bevor eine Iteration, einschließlich der ersten Iteration, ausgeführt wird.

Do-While-Schleife

Der Operator "Schleife" überprüft am Ende jeder Iteration die Schleifenbedingung, die im Feld "Wiederholungsanzahl" angegeben ist, um sicherzustellen, dass immer mindestens die erste Iteration der Schleife ausgeführt wird.

Hinweis: In allen vorhandenen "Schleifen"-Operatoren, die aus CA Process Automation, vor der Version v4, importiert wurden, ist das Feld "While-Schleife" deaktiviert. Wie in den Vorgängerversionen funktionieren diese vorhandenen Operatoren weiterhin als Do-While-Schleifen.

Vor- und Nachbearbeitungscode

Verwenden Sie diese Felder, um einen JavaScript-Code auszuführen, der mit jeder Iteration der Schleife ausgeführt werden soll.

Die Verarbeitungssequenz von einer Vor- und Nachbedingung hängt vom Typ von Schleife ab. Im *Handbuch für Inhaltsdesign* erfahren Sie mehr über die logische Sequenz einer Schleife.

Ausgabeparameter

CurrentLoopIteration

Ein Schleifenzähler, der während der ersten Iteration bei 0 beginnt und für jede zusätzliche Iteration um 1 erhöht wird. Diese Variable am Anfang und am Ende jeder Iteration aktualisiert. Wenn der Operator "Schleife" so konfiguriert ist, dass er dreimal in einer Schleife ausgeführt wird, ist "CurrentLoopIteration" am Ende der Ausführung aller Iterationen gleich:

- 0 in Iteration 1
- 1 in Iteration 2
- 2 in Iteration 3
- 3 in der letzten Iteration, die nicht ausgeführt wird, da sie die Bedingung des Operators "Schleife" verletzt.

OverallLoopDuration

Ein Schleifenzähler, der die Zeit (in Sekunden) angibt, die seit dem Start der ersten Iteration der Schleife vergangen ist. Diese Variable wird am Anfang und Ende jeder Iteration aktualisiert und schließt Verzögerungen ein, die zwischen Iterationen der Schleife festgelegt sind.

Manuelles Zurücksetzen des Operators "Schleife" in einem Prozess

Im *Handbuch für Inhaltsdesign* finden Sie weitere Informationen über das manuelle Zurücksetzen des Operators "Schleife" in einem Prozess.

Operator "Prozessspur ändern"

Der Operator "Prozessspurwechsel" initiiert eine Reihe von Prozessspurwechselregeln im Bereich "Prozessspurwechsel-Handler" des Designers.

So fügen Sie den Operator "Prozessspurwechsel" in einen Prozess ein

Ziehen Sie den Operator "Prozessspur ändern" vom Standardordner in einen der Editoren.

Eingabeparameter

Name

Diese Option zeigt den Namen des Prozessspurwechsels an. Sie können den Namen durch das Bearbeiten der Eigenschaft "Name" unter der Eigenschaftsgruppe "Informationen" für den Prozessspurwechsel ändern.

Quelle

Gibt die Quellprozessspur für den Prozessspurwechsel an. Wählen Sie "All" für einen Prozessspurwechsel aus einer Prozessspur aus.

Ziel

Gibt die Zielprozessspur für den Prozessspurwechsel an. Wählen Sie "Alle" für einen Prozessspurwechsel in eine Prozessspur aus.

Operator "Ausnahme"

Verwenden Sie den Operator "Ausnahme", um eine Ausnahme zu initiieren, beispielsweise eine Beendigung aufgrund von Systemfehlern oder nicht identifizierten Beendigungsbedingungen. Um den Operator "Ausnahme" in einen Prozess einzuordnen, ziehen Sie ihn aus dem Standardordner in den Ausnahmen-Handler-Editor.

Eingabeparameter

Name

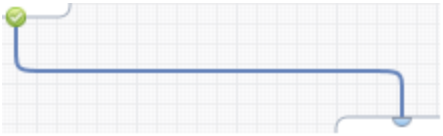
Zeigt den Namen der Ausnahme an. Um den Namen zu ändern, bearbeiten Sie unter der Eigenschaftsgruppe "Informationen" die Eigenschaft "Name" für die Ausnahme.

Ausnahmetyp

Wählen Sie aus der Dropdown-Liste "Systemfehler", "Nicht identifizierte Antwort", "Abgebrochen" oder "Zeitlimit" aus, um die Ausnahme zu kategorisieren.

Links

Verknüpfungen definieren die Struktur eines Prozesses, indem Sequenzen von Operatoren erstellt werden.



So erstellen Sie eine Verknüpfung

Klicken Sie auf einem Operator auf eine Ausgangsverknüpfung, und ziehen Sie sie zum nachfolgenden Operator in der Sequenz.

Wenn die gewünschte Verknüpfung nicht angezeigt wird, klicken Sie mit der rechten Maustaste auf den Operator und dann im Kontextmenü auf den Verknüpfungstyp (wie "Fehlgeschlagen", "Abgeschlossen" oder "Anwenderspezifisch").

Verknüpfungseigenschaften

Verknüpfungseigenschaften werden angezeigt, wenn Sie mit der rechten Maustaste auf eine Verknüpfung in einem Prozess klicken und dann "Eigenschaften" wählen.

Gewichtung

Gibt die Dicke der Linien zwischen Operatoren an.

Farbe

Öffnet das Dialogfeld "Verknüpfungsfarbe auswählen", in dem Sie die Farbe von Verknüpfungen im Prozess ändern können.

Formen

Gibt die Linienform für Verknüpfungen zwischen Operatoren an:

Gerade

Erstellt gerade Verknüpfungen zwischen Operatoren.

Rechtwinklig

Erstellt rechtwinklige Verknüpfungen zwischen Operatoren.

Gestrichelt

Aktivieren Sie dieses Kontrollkästchen, um eine gestrichelte (punktiert) Verknüpfung zu erstellen.

Kapitel 3: Catalyst

Die Catalyst-Operatoren schließen das Erstellen, Lesen, Aktualisieren und Löschen und Operatoren für Event-Abonnements ein, die auf einem beliebigen Catalyst-Connector abgerufen werden können. Alle Parameter für den Operator können Ausdrücke für größtmögliche Flexibilität im Erstellungsinhalt enthalten. CA Process Automation-Prozesse können mithilfe einer Kombination dieser Operatoren erstellt werden, um Integrationen in mehreren Produkten zu erstellen. Außerdem können die Catalyst-Operatoren als Basisoperatoren verwendet werden, um anwenderspezifische Operatoren für produktspezifische Lösungen zu erstellen.

Alle Catalyst-Connectors stellen Objekte aus, die dem vereinheitlichten Servicemodell (Unified Service Model, USM) entsprechen. Dieses gewöhnliche Modell erleichtert übergreifende Produktintegrationen.

Die Catalyst-Operatoren enthalten Operatoren, die für jeden USM-Typ generisch sind. Diese Operatoren enthalten auch Operatoren, die für jeden USM-Typ spezifisch sind. Informationen finden Sie im Handbuch zum Connector, das Sie mit dem anwendbaren Catalyst-Connector erhalten.

Catalyst-Knoten enthalten einen Broker, der ein Verzeichnis von Connectors ist. Die Catalyst-Operatoren fragen im Designmodus den Broker ab und zeigen die Connectornamen in der MdrProdInstance-Liste an.

Generische USM-Operatoren

Es folgen allgemein verwendeten Operatoren, die für alle USM-Typen gelten.

Operator "Erstellen"

Der Operator "Erstellen" unterstützt die CRUD-Erstellungs- und Aktualisierungsvorgänge für alle USM-Typen.

Eingabeparameter

UCFBrokerURL

Definiert die UCF-Broker-URL Ihres Catalyst-Servers. Standardmäßig die UCF-Broker-URL in der Catalyst-Konfiguration.

Geben Sie die Broker-URL des Catalyst-Broker-Service wie folgt an:

`http://<hostname>:7000/ucf/BrokerService`

Geben Sie bei Verwendung sicherer Catalyst-Kommunikationen die sichere Broker-URL wie folgt an:

`https://<hostname>:7443/ucf/BrokerService`

MdrProduct

Eindeutige Kennung des verbundenen Produkts.

MdrProdInstance

Eindeutige Kennung der Instanz des verbundenen Produkts, wie im UCF-Broker registriert. CA Process Automation fragt den UCF-Broker nach der Liste verfügbarer Connectors ab und füllt dieses Feld auf. Sie können dann Ihren Connector aus der Drop-down-Liste auswählen.

Erstellen

Zeigt an, ob der Operator:

- Ein Objekt erstellt

  ODER

- Ein vorhandenes Objekt aktualisiert.

Itemtype

Gibt den USM-Typ des Objekts an, das erstellt oder aktualisiert wird.

Werte:

Alarm, ComputerSystem, Router, Service usw.

Beispiel:

`itemtype=ComputerSystem`

Eigenschaften

Die Parameter für den Operator enthalten die Eigenschaften des USM-Typs.

Anpassen der Eigenschaften

Das Formular "Eigenschaften" kann mithilfe von "Name der Produkteigenschaften-Konfigurationsdatei" angepasst werden. Wenn die Werte "MdrProduct" und "Itemtype" mit einer Eingabe in "Name der Produkteigenschaften-Konfigurationsdatei" übereinstimmen, wird das Formular entsprechend den folgenden Regeln angezeigt:

- Wenn die Eigenschaft nicht im USM-Typ definiert wird, ist es eine anwenderspezifische Eigenschaft und wird dem Formular hinzugefügt.
- Wenn die Eigenschaft im USM-Typ definiert wird, dann wird sie dem Formular hinzugefügt.
- Wenn die Eigenschaft im USM-Typ definiert wird und über einen Aliasnamen verfügt, dann wird sie mit dem Aliasnamen zum Formular hinzugefügt.

  Zum Beispiel zeigt diese Eingabe das Alarmformular folgendermaßen an.

```
<!-- SCOM  -->
<MdrTypes MdrProduct="CA:00031">
    <TypeMap name="Alert">
      <Mapping propName="MdrProdInstance" aliasName="siloHost" />
      <Mapping propName="MdrElementId" aliasName="Id" />
      <Mapping propName="UrlParams" aliasName="" />
      <Mapping propName="SeverityTrend" aliasName="" />
      <Mapping propName="RelatedAlerts" aliasName="" />
      <Mapping propName="AlertedMdrProdInstance" aliasName="siloHost" />
      <Mapping propName="AlertedMdrElementID" aliasName="MonitoringObjectId"
/>
      <Mapping propName="Summary" aliasName="Name" />
      <Mapping propName="Message" aliasName="Description" />
      <Mapping propName="Assignee" aliasName="Owner" />
    </TypeMap>

</MdrTypes>
```

  Die QuickInfo der Eigenschaften mit Aliasnamen zeigt den USM-Eigenschaftsnamen an.

Operator "Löschen"

Der Operator "Löschen" unterstützt den CRUD-Löschvorgang für alle USM-Typen. Die Parameter identifizieren das MDR und das zu löschende Objekt.

Eingabeparameter

UCFBrokerURL

Definiert die UCF-Broker-URL Ihres Catalyst-Servers. Standardmäßig die UCF-Broker-URL in der Catalyst-Konfiguration.

Geben Sie die Broker-URL des Catalyst-Broker-Service wie folgt an:

`http://<hostname>:7000/ucf/BrokerService`

Geben Sie bei Verwendung sicherer Catalyst-Kommunikationen die sichere Broker-URL wie folgt an:

`https://<hostname>:7443/ucf/BrokerService`

MdrProduct

Eindeutige Kennung des verbundenen Produkts.

MdrProdInstance

Eindeutige Kennung der Instanz des verbundenen Produkts, wie im UCF-Broker registriert. CA Process Automation fragt den UCF-Broker nach der Liste verfügbarer Connectors ab und füllt dieses Feld auf. Sie können dann Ihren Connector aus der Drop-down-Liste auswählen.

MdrElementID

Eindeutige Kennung des Objekts im verbundenen Produkt.

ClassName

Klassenname des Objekts (Alert, ComputerSystem usw.)

Operator "Ausführen"

Der Operator "Ausführen" unterstützt anwenderspezifische Betriebsabläufe auf einem beliebigen UCF-Connector. Die Parameter identifizieren das MDR, den Betriebsablauf und die Betriebsablaufparameter.

Eingabeparameter

CatalystBrokerURL

Definiert die UCF-Broker-URL Ihres Catalyst-Servers. Standardmäßig ist dieser Wert die UCF-Broker-URL der Catalyst-Konfiguration.

Geben Sie die Broker-URL des Catalyst-Broker-Service wie folgt an:

`"http://hostname:7000/ucf/BrokerService"`

Geben Sie bei Verwendung sicherer Catalyst-Kommunikationen die sichere Broker-URL wie folgt an:

`"https://hostname:7443/ucf/BrokerService"`

MdrProduct

Definiert eine eindeutige Kennung des verbundenen Produkts.

CA Process Automation wird folgendermaßen identifiziert:

```
"CA:00074 (CA Process Automation)"
```

MdrProdInstance

Definiert eine eindeutige Kennung der Instanz des verbundenen Produkts entsprechend der Registrierung im UCF-Broker. CA Process Automation fragt die Liste verfügbarer Connectors vom Catalyst-Broker ab und füllt dieses Feld damit auf, sodass Sie Ihren Connector aus der Drop-down-Liste auswählen können.

CA Process Automation wird folgendermaßen identifiziert:

```
"CA:00074:01"
```

Betriebsablaufskategorie

Gibt die geladenen Connector-Deskriptoren für die Catalyst-Operatoren an. Wählen Sie einen Deskriptor aus der Drop-down-Liste aus.

Operation

Gibt die Vorgänge des Connector-Deskriptors an, nachdem Sie eine Betriebsablaufskategorie ausgewählt haben. Wählen Sie einen Connector-Deskriptor aus der Drop-down-Liste aus.

ParameterExpression

Definiert ein ValueMap, das mit der erwarteten Parameterstruktur übereinstimmt.

Verwenden Sie diesen Operator, wenn der Operator "Ausführen" als Basisoperator eines anwenderspezifischen Operators verwendet wird und der Vorbearbeitungscode den Wert dynamisch erstellt. Sie können den Datensatz des Basisoperators "Ausführen" als Referenz verwenden, um den Wert einzurichten.

ParameterNamespaceExpression

Definiert die Namespaces, die im Parameterausdruck verwendet werden.

Verwenden Sie diesen Operator, wenn der Operator "Ausführen" als Basisoperator eines anwenderspezifischen Operators verwendet wird und der Vorbearbeitungscode den Wert dynamisch erstellt. Sie können den Datensatz des Basisoperators "Ausführen" als Referenz verwenden, um den Wert einzurichten.

Parameter

Wenn Sie einen Betriebsablauf ausgewählt haben, klicken Sie auf "Parameter", um die Eingabeparameter für die Abfrage anzugeben. Diese Felder variieren je nach dem ausgewählten Betriebsablauf. Geben Sie keine Daten ein, wenn Sie in den Feldern für Parameter-Ausdruck und Parameter-Namespace bereits Daten eingegeben haben.

Parameter für die einzelnen Betriebsabläufe finden Sie hier.

Parameter von Operation-Specific

Geben Sie die folgenden Daten ein, um Informationen zum ausgewählten Betriebsablauf anzugeben.

Hinweis: Geben Sie keine Daten ein, wenn Sie in den Feldern für Parameter-Ausdruck und Parameter-Namespace bereits Daten eingegeben haben.

AbortStartRequest

Instanz

Gibt die Instanz des Startauftrags an, der abgebrochen werden soll.

Abbrechen

MdrProduct

Gibt die eindeutige Kennung des verbindenden Produkts an, mit dem Sie den Auftrag abbrechen wollen.

MdrProdInstance

Gibt die eindeutige Kennung der Instanz des verbindenden Produkts (gemäß der Registrierung im Catalyst-Broker) an, mit dem Sie den Auftrag abbrechen wollen.

MdrElementID

DelegateTask

TaskID

Gibt die ID der Aufgabe an, die Sie abbrechen wollen.

Exportieren

Gibt Informationen zum exportierten Inhalt an.

ExportRequest

- Ordnername: Gibt den Namen des Ordners an, der den exportierten Inhalt enthält.
- IsAbsolute:
- SealModifiableReleaseVersions:

ExportFileName

Gibt an:

- ExportLocation
- OverwriteFile

Filter

ObjectTypes: Ermöglicht es Ihnen, nach einem bestimmten Objekttyp zu filtern.

Halten

MdrProduct

Gibt die eindeutige Kennung des verbindenden Produkts an, mit dem Sie

MdrProdInstance

Gibt die eindeutige Kennung der Instanz des verbindenden Produkts (gemäß der Registrierung im Catalyst-Broker) an, mit dem Sie

MdrElementID

Importieren

ImportLocation

Gibt den Speicherort der Textdatei an, die importiert werden soll.

SourceLocation

OverwriteAction

SetCurrent

MakeAvailable

QueryDatasetNames

PathName

Gibt den Pfad an, der den Namen des Datensatzes enthält, für den Sie eine Abfrage durchführen wollen.

QueryDatasetParameters

PathName

Gibt den Pfad an, der den Namen des Datensatzes enthält, für den Sie eine Abfrage durchführen wollen.

QueryModuleConfigProperties

ModuleName

Gibt den Namen der CA Process Automation-Moduleigenschaften an, die Sie konfigurieren wollen.

QueryModuleConfigs

(Optional) ModuleName

Gibt den Namen des CA Process Automation-Moduls an.

(Optional) <ModuleType>

Gibt an, ob die standardmäßige Modulkonfiguration oder eine anwenderspezifische Modulgruppenkonfiguration zurückgegeben wird. Standardmäßig gibt die Methode "QueryModuleConfigs" alle verfügbaren Module zurück. Das Attribut <ModuleType> übernimmt die folgenden Werte:

Alle

Gibt die standardmäßige Modulkonfiguration und die anwenderspezifische Modulkonfiguration zurück.

Standard

Gibt nur die standardmäßige Modulkonfiguration zurück.

Anwenderdefiniert

Gibt nur die anwenderspezifische Modulkonfiguration zurück.

QueryStartRequestForms

LookUpPath

Gibt an:

IsRecursive

Werte

Der Wert kann einer der folgenden sein:

- wahr: Geben Sie an "wahr" an, um
- falsch: Geben Sie an "falsch" an, um

Schlüsselwörter

Fügen Sie einen Parameter hinzu...

QueryStartRequests

Instanz

Gibt die Startauftragsinstanz an, die abgefragt werden soll.

IsArchived

Gibt an, ob die Startauftragsinstanz archiviert ist.

Der Wert kann einer der folgenden sein:

- wahr: CA Process Automation überprüft, ob die angegebene Startauftragsinstanz in der Datenbank mit archivierten Tabellen verfügbar ist. Wenn sich die Startauftragsinstanz nicht in der Datenbank mit archivierten Tabellen befindet, gibt der angegebene Instanzenname alle verfügbaren archivierten Instanzen aus der Datenbank zurück.

- falsch: CA Process Automation überprüft, ob die angegebene Startauftragsinstanz in der Datenbank mit aktuellen Tabellen verfügbar ist. Wenn sich die Startauftragsinstanz nicht in der Datenbank mit aktuellen Tabellen befindet, gibt der angegebene Instanzenname alle verfügbaren archivierten Instanzen aus der Datenbank zurück.

QueryTasks

TaskID

Gibt die ID der Aufgabe an, die Sie abfragen wollen.

RootUUID

ProcessID

Zustand

Version

MdrProduct

Gibt die eindeutige Kennung des verbindenden Produkts an, mit dem Sie

MdrProdInstance

Gibt die eindeutige Kennung der Instanz des verbindenden Produkts (gemäß der Registrierung im Catalyst-Broker) an, mit dem Sie

MdrElementID

ReplyTask

TaskID

Gibt die ID der Aufgabe an, auf die Sie antworten wollen.

IsApproved

Gibt die ID der Aufgabe an, die Sie genehmigen wollen.

Argumente

ReturnTask

TaskID

Gibt die ID der Aufgabe an, die Sie zurückgeben wollen.

Start

MdrProduct

Gibt die eindeutige Kennung des verbindenden Produkts an, mit dem Sie

MdrProdInstance

Gibt die eindeutige Kennung der Instanz des verbindenden Produkts (gemäß der Registrierung im Catalyst-Broker) an, mit dem Sie

MdrElementID

Argument

SubmitStartRequestForm

Name

Gibt den Namen des Startauftragsformulars an, das gesendet werden soll.

Path

Gibt den Pfad des Startauftragsformulars an, das gesendet werden soll.

Argumente

StartDate

Gibt das Startdatum des Startauftragsformulars an, das gesendet werden soll.

StartTime

Gibt den Zeitpunkt an, zu dem das Startauftragsformular, das Sie senden wollen, angefangen hat.

Priorität

Gibt die Priorität des Startauftragsformulars, das Sie senden wollen, an.

TakeTask

TaskID

Gibt die ID der Aufgabe an, die Sie

TransferTask

TaskID

Gibt die ID der Aufgabe an, die Sie

Anwender

Gruppen

UpdateDatasetParameters

PathName

Gibt den Namen des Pfades für den Datensatz an, der die Parameter enthält, die Sie aktualisieren wollen.

Parameter

Gibt einen Parameter für den Datensatz an, den Sie aktualisieren wollen. ?? Gibt dies den neuen Wert an?

Operator "Abrufen"

Der Operator "Abrufen" unterstützt den CRUD-Lesevorgang für alle USM-Typen. Die Parameter identifizieren das MDR und die UCF-Filterwerte (entitytype, itemtype, rekursiv, id und updatedAfter). Außerdem beschränkt der Parameter "MaxNumberOfObjects" die Anzahl der Objekte, die vom Operator zurückgegeben werden.

Eingabeparameter

UCFBrokerURL

Definiert die UCF-Broker-URL Ihres Catalyst-Servers. Standardmäßig die UCF-Broker-URL in der Catalyst-Konfiguration.

Geben Sie die Broker-URL des Catalyst-Broker-Service wie folgt an:

`http://<hostname>:7000/ucf/BrokerService`

Geben Sie bei Verwendung sicherer Catalyst-Kommunikationen die sichere Broker-URL wie folgt an:

`https://<hostname>:7443/ucf/BrokerService`

MdrProduct

Eindeutige Kennung des verbundenen Produkts.

MdrProdInstance

Eindeutige Kennung der Instanz des verbundenen Produkts, wie im UCF-Broker registriert. CA Process Automation fragt den UCF-Broker nach der Liste verfügbarer Connectors ab und füllt dieses Feld auf. Sie können dann Ihren Connector aus der Drop-down-Liste auswählen.

entitytype

Gibt den Typ der Entität an. Werte können "Alert", Item" oder "Relationship" sein.

Beispiel:

```
entitytype=Item
```

itemtype

Gibt den Typ des Elements an. Wenn nicht angegeben, werden alle Typen abgerufen.

Beispiel:

```
itemtype=ComputerSystem
```

rekursiv

Gibt an, ob der Connector das Element und die untergeordneten Objekte und Beziehungen, aus denen es besteht, rekursiv einschließen soll.

id

Gibt eine bestimmte Objekt-ID an (entspricht "MdrElementID")

updatedAfter

Gibt an, dass nur Objekte nach einer bestimmten Zeit aktualisiert wurden.

MaxNumberOfObjects

Gibt die Höchstanzahl an Objekten an, die abgerufen werden, bevor der Operator abgeschlossen wird.

Operator "Änderungen abonnieren"

Der Operator "Änderungen abonnieren" unterstützt Event-Abonnements auf einem USM-Typ mithilfe von UCF-Filtern. Die Parameter identifizieren das MDR und die UCF-Filterwerte (entitytype, itemtype, rekursiv, id und updatedAfter). Außerdem beschränkt der Parameter "MaxNumberOfObjects" die Anzahl der Objekte, die vom Operator zurückgegeben werden. Der Parameter für "TimeOut" gibt die Anzahl der Sekunden an, nach denen das Abonnement abläuft. Der Operator wird abgeschlossen, wenn entweder die Anzahl der Objekte zurückgegeben werden oder wenn das Zeitlimit abläuft.

Eingabeparameter

Dieser Operator übernimmt folgende Eingabeparameter:

UCFBrokerURL

Definiert die UCF-Broker-URL Ihres Catalyst-Servers. Standardmäßig die UCF-Broker-URL in der Catalyst-Konfiguration.

Geben Sie die Broker-URL des Catalyst-Broker-Service wie folgt an:

`http://<hostname>:7000/ucf/BrokerService`

Geben Sie bei Verwendung sicherer Catalyst-Kommunikationen die sichere Broker-URL wie folgt an:

`https://<hostname>:7443/ucf/BrokerService`

MdrProduct

Eindeutige Kennung des verbundenen Produkts.

MdrProdInstance

Eindeutige Kennung der Instanz des verbundenen Produkts, wie im UCF-Broker registriert. CA Process Automation fragt den UCF-Broker nach der Liste verfügbarer Connectors ab und füllt dieses Feld auf. Sie können dann Ihren Connector aus der Drop-down-Liste auswählen.

entitytype

Gibt den Typ der Entität an. Werte können "Alert", Item" oder "Relationship" sein.

Beispiel:

`entitytype=Item`

itemtype

Gibt den Typ des Elements an. Wenn nicht angegeben, werden alle Typen abgerufen.

Beispiel:

`itemtype=ComputerSystem`

rekursiv

Gibt an, ob der Connector das Element und die untergeordneten Objekte und Beziehungen, aus denen es besteht, rekursiv einschließen soll.

id

Gibt eine bestimmte Objekt-ID an (entspricht "MdrElementID").

updatedAfter

Gibt an, dass nur Objekte nach einer bestimmten Zeit aktualisiert wurden.

timeOut

Gibt die Anzahl der Sekunden an, nach denen das Abonnement abläuft.

MaxNumberOfObjects

Gibt die Höchstanzahl an Objekten an, die abgerufen werden, bevor der Operator abgeschlossen wird.

Beispiel

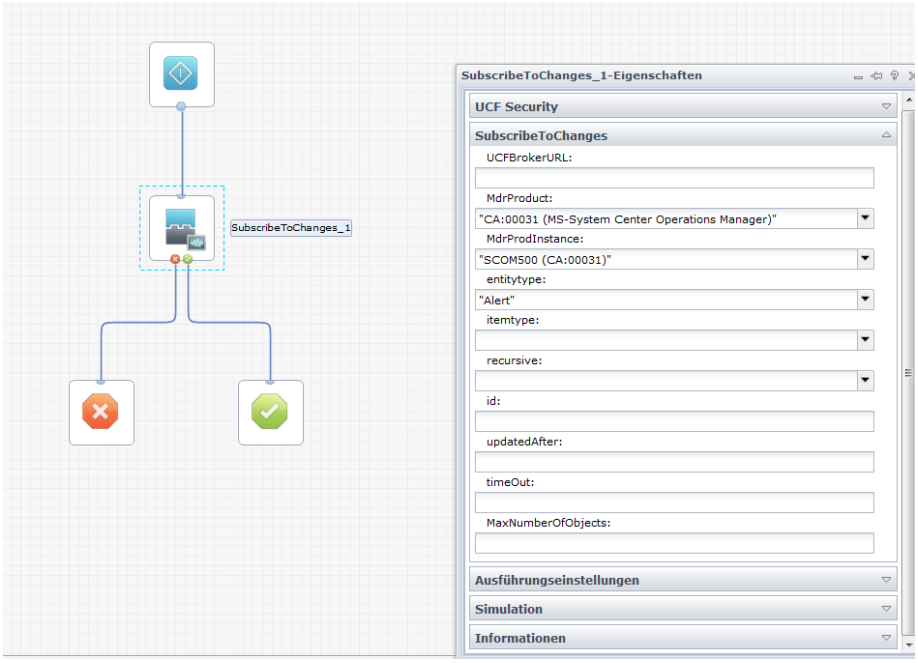
Die Catalyst-Operatoren können direkt in Prozessen verwendet werden, um generischen Inhalt zu erstellen. Sie können auch die Operatoren als Basisoperatoren von anwenderspezifischen Operatoren für produktspezifischen Inhalt verwenden.

In diesem Beispiel wird beschrieben, wie Service Desk Incidents von SCOM-Alarmen erstellt werden.

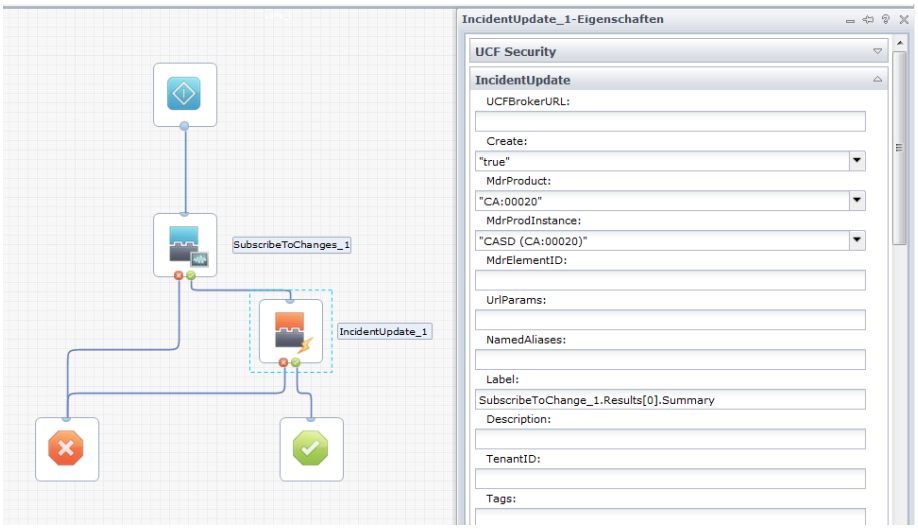
So erstellen Sie eine einfache Lösung

1. Fügen Sie einem Prozess den Operator "Änderungen abonnieren" hinzu.
2. Wählen Sie den SCOM-Connector aus den MdrProduct/MdrProdInstance-Listen und Alarm von der Entitytype-Liste aus.

3. Fügen Sie dem Prozess den Operator "Incident-Update" hinzu. Aktivieren Sie das Kontrollkästchen "Erstellen" und "Service Desk-Connector" aus der Liste.



4. Legen Sie die Werte der Incidents-Eigenschaften mithilfe der Eigenschaften des SCOM-Alarmobjekts als variable Ausdrücke fest.



Der Inhalt kann nun verwendet werden.

Nicht generische USM-Operatoren

Zusätzlich zu den generischen CRUD-Operatoren, gibt es für jeden USM-Typ spezifische Operatoren zum Erstellen und Aktualisieren. Die Parameter der Operatoren werden aus den Eigenschaften der USM-Typen erstellt. Diese Operatoren werden dynamisch vom USM-Schema während der Initialisierung der Catalyst-Initialisierung erstellt.

Folgende Operatoren wurden speziell aus USM-Typen generiert:

- AlertUpdate
- ApplicationServerUpdate
- ApplicationSystemUpdate
- ApplicationUpdate
- AssetUpdate
- BackgroundProcessUpdate
- BinaryRelationshipsUpdate
- BootSoftwareUpdate
- BusinessProcessServerUpdate
- BusinessTransactionUpdate
- ChangeOrderUpdate
- ChangePackageUpdate
- ClusterUpdate
- CommentUpdate
- CommunicationServerUpdate
- ComplianceStatusUpdate
- ComputerSystemUpdate
- ConnectorIDUpdate
- ConnectorUpdate
- ContractUpdate
- DatabaseInstanceUpdate
- DatabaseUpdate
- DirectoryServerUpdate
- DiskPartitionUpdate
- EntityIDUpdate
- EntityUpdate
- EnvironmentSensorUpdate

- ExtensionEntityUpdate
- ExtensionRunningHardwareUpdate
- FileUpdate
- GenericIPDeviceUpdate
- GroupUpdate
- HypervisorManagerUpdate
- IncidentUpdate
- InterfaceCardUpdate
- IPConfigUpdate
- ITActivityProfileUpdate
- ITActivityTemplateUpdate
- ITActivityUpdate
- LatestUsmBuildUpdate
- LocationUpdate
- MailServerUpdate
- ManagedAccesssUpdate
- ManagementAgentUpdate
- MediaDriveUpdate
- MemoryUpdate
- MessageServerUpdate
- MultiFunctionEntityUpdate
- NetworkServerUpdate
- NetworkUpdate
- OperatingSystemUpdate
- OrganizationalEntityUpdate
- PersonUpdate
- PhysicalContainerUpdate
- PortUpdate
- PowerSupplyUpdate
- PrinterUpdate
- PrintServerUpdate
- ProblemUpdate
- ProcessorUpdate

- ProjectUpdate
- ProvisionedSoftwareUpdate
- RequestUpdate
- ResourceServerUpdate
- RouterUpdate
- RunningHardwareUpdate
- RunningSoftwareUpdate
- SecurityServerUpdate
- ServiceSpecificationUpdate
- ServiceUpdate
- SnmpV1AccessUpdate
- SnmpV3AccessUpdate
- SoftwareComponentUpdate
- StorageArrayUpdate
- StoragePoolUpdate
- StorageVolumeUpdate
- SwitchUpdate
- TablespaceUpdate
- TransactionContextUpdate
- TransactionSegmentUpdate
- TransactionServerUpdate
- VirtualizationManagerUpdate
- VirtualSystemUpdate
- VMDataStoreUpdate

Catalyst-Sicherheitsparameter

Jeder Catalyst-Operator enthält Catalyst-Sicherheitsparameter. Diese Parameter unterstützen die Authentifizierung sowohl auf Catalyst- als auch auf Connector-Ebene.

Nachdem der Zugriff auf Catalyst-Knoten gewährt wurde, können Sie Ansprüche verwenden, um connectorspezifische Sicherheitsinformationen abzurufen. Informationen über connectorspezifische Ansprüche finden Sie im Handbuch zum Connector, das Sie mit dem anwendbaren Catalyst-Connector erhalten.

Anwendername

Definiert die Anwender-ID, die auf Catalyst-Knoten zugreift.

Kennwort

Definiert das Kennwort, das mit dem Anwendernamen verbunden ist.

Da Sie das Kennwort als Ausdruck angeben, ist der Text, den Sie eingeben, sichtbar. Vermeiden Sie wörtliche Zeichenfolgen und geben Sie einen Ausdruck ein, der sich auf eine Kennwortvariable in einem globalen Datensatz bezieht.

Ansprüche

Diese Ansprüche sind nicht kennwortgeschützt.

Klicken Sie auf "Hinzufügen". Geben Sie den ersten Anspruchsnamen und dessen Wert ein. Wiederholen Sie diesen Schritt für jeden Anspruch. Verwenden Sie die Auf- und Abwärtspfeile, um die Ansprüche sequenziell zu ordnen oder nach Bedarf zu löschen.

Anspruchsname

Definiert den Namen des Anspruchs.

Anspruchswert

Definiert den Wert des benannten Anspruchs.

Kennwortansprüche

Diese Ansprüche sind kennwortgeschützt. CA Process Automation verschlüsselt die Kennwortwerte.

Klicken Sie auf "Hinzufügen", und geben Sie den ersten Kennwortanspruchsnamen und dessen Wert ein. Wiederholen Sie diesen Schritt für jeden Kennwortanspruch. Verwenden Sie die Auf- und Abwärtspfeile, um die Ansprüche sequenziell zu ordnen oder nach Bedarf zu löschen.

Anspruchsname

Definiert den Namen des Anspruchs.

Anspruchswert

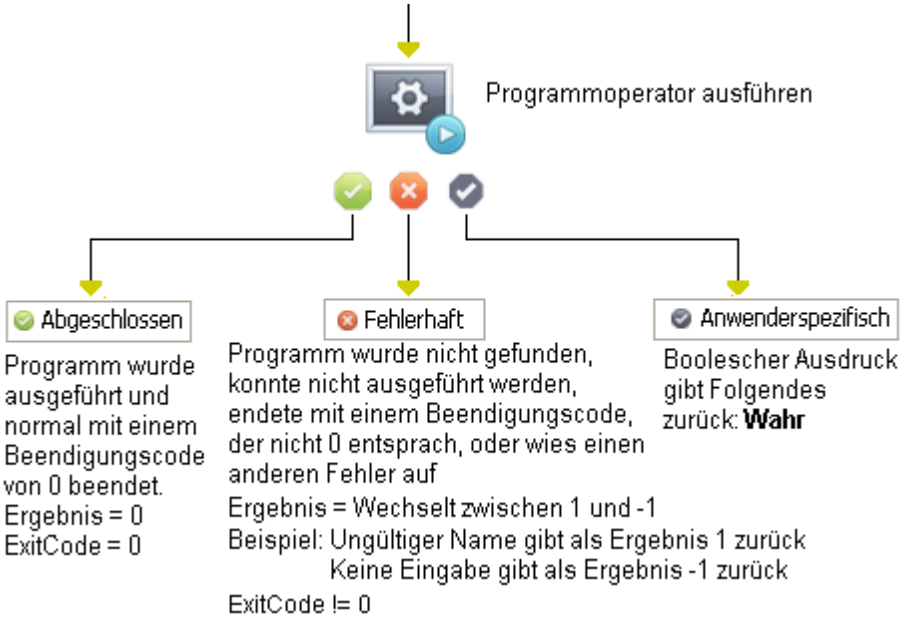
Definiert den Wert des benannten Anspruchs.

Kapitel 4: Befehls-Ausführung

Die Befehlsausführungs-Operatoren führen Prozesse und Skripte in der Host-Betriebsumgebung aus.

Operator "Programm ausführen"

Der Operator "Programm ausführen" startet und überwacht ein Programm.


Programmoperator ausführen
Abgeschlossen
Programm wurde ausgeführt und normal mit einem Beendigungscode von 0 beendet.
Ergebnis = 0
ExitCode = 0
Fehlerhaft
Programm wurde nicht gefunden, konnte nicht ausgeführt werden, endete mit einem Beendigungscode, der nicht 0 entsprach, oder wies einen anderen Fehler auf
Ergebnis = Wechselt zwischen 1 und -1
Beispiel: Ungültiger Name gibt als Ergebnis 1 zurück
Keine Eingabe gibt als Ergebnis -1 zurück
ExitCode != 0
Anwenderspezifisch
Boolescher Ausdruck gibt Folgendes zurück: Wahr

Hinweis: Für fast alle Operatoren wird "Ergebnis" bei erfolgreicher Ausführung auf "1" festgelegt und auf eine andere Zahl, die nicht "1" entspricht, bei einer fehlerhaften Ausführung. Für diesen bestimmten Operator wird "Ergebnis" bei erfolgreicher Ausführung auf "0" festgelegt und auf eine andere Zahl, die nicht "0" entspricht, bei einer fehlerhaften Ausführung.

Eingabeparameter

Programmname

- UNIX-Ziel: Das Programm muss entweder eine binäre Datei oder ein Shell-Skript sein, das den UNIX Konventionen entspricht (die erste Zeile der Datei muss den vollständigen Pfad der Shell enthalten, wie in #!/bin/ksh).
- Windows-Ziel: Das Programm muss eine ausführbare Datei oder ein Skript sein. Die Datei wird normalerweise angegeben durch:
  - Den Namen der im Arbeitsverzeichnis für den Operator auszuführenden Datei

    ODER
  - In einem der von der Umgebungsvariable des Pfads auf dem Zielhost angegebenen Verzeichnisse.

Der Pfad, der relativ zu dem Arbeitsverzeichnis ist, das für die Operatoren "Eigenschaften der UNIX-Befehlsausführung" angegeben ist.

Profil

UNIX

Gibt ein Hostsystemprofil an, das als "Quelle" für die Definition von Teilen des Ausführungszusammenhangs des Prozesses verwendet werden soll.

Beispiel:

```
"/home/anwendername/appli_1_profil"
```

Wenn kein Profil hier angegeben wird, wird das in den Eigenschaftseinstellungen von "Eigenschaften der UNIX-Befehlsausführung" angegebene Profil (auf Kategorienebene) verwendet. Ist weder im Operator noch in den Operator-Einstellungen ein Profil angegeben, wird nur das Anwenderprofil verwendet, mit dem der Prozess ausgeführt wird.

Windows

Gibt den Pfad zu einer Batch-Datei (.bat) in Windows an, in der Umgebungsvariablen festgelegt werden, die im Prozess verwendet werden sollen.

Die Variableneinstellungen in der hier angegebenen Batch-Datei gibt es zusätzlich zu allen Variableneinstellungen, die für das Anwenderprofil definiert sind, mit dem der Prozess ausgeführt wird. Wenn für diese Option keine Datei angegeben ist, verwendet der Operator die Einstellungen, die durch den Shell-Profilparameter definiert sind, der für den Windows-Prozess-Service festgelegt ist.

Arbeitsverzeichnis

Normalerweise gibt dieses Feld den Ordner an, der die Programmdatei oder einige zugehörige Dateien enthält, die das Programm benötigt. Jede Datei, die ohne einen ausdrücklichen Pfad angegeben ist, wird in diesem Verzeichnis erstellt oder gesucht.

Wenn Sie kein Arbeitsverzeichnis angeben, wird standardmäßig das Start- oder Arbeitsverzeichnis des Anwenderkontos verwendet, unter dem der Operator ausgeführt wird.

UNIX

Gibt das Arbeitsverzeichnis für den Operator an.

Zum Beispiel: "/home/anwender1"

Anwender-ID

Gibt den Systemanwendernamen an, unter dem das Programm ausgeführt werden soll. Der Anwender muss Ausführberechtigungen für die Datei haben.

Wenn Sie nichts in dieses Feld eingeben, ist die Standardeinstellung der Standardanwender, der in den Konfigurationseinstellungen auf Kategorieebene angegeben ist.

Anwendernamen und Kennwörter können auf Kategorienebene angegeben oder in benannten Datensatzvariablen gespeichert werden. Auf diese Weise können sie zentral aktualisiert werden, ohne dabei Prozesswerte zu ändern.

Kennwort

Gibt das aktuelle Kennwort für die angegebene Anwender-ID an.

Das Kennwort als Literalzeichenfolge anzugeben, ist keine empfohlene Methode. Vielmehr empfiehlt es sich, das Kennwort in einer Datensatzvariable vom Typ "Kennwort" zu verwenden und die Variable zu übergeben.

Parameter

Die Parameter, die an den Prozess übergeben werden sollen.

Die Parameter werden in der Reihenfolge an den Prozess übergeben, in der sie hier aufgelistet sind. Verwenden Sie die Schaltflächen, um Parameter hinzuzufügen, zu entfernen oder neu anzuordnen.

Programmparameter werden beim Start einzeln an das Programm übergeben.

Standardmäßige Ausgangsdatei

Gibt die Datei an, um vom Programm in STDOUT geschriebenen Text zu erfassen.

Beispiel:

```
/tmp/trace.log
```

Sie können sowohl für die standardmäßige Fehler- als auch die standardmäßige Ausgabedatei dieselbe Datei angeben. Es wird allerdings keine Reihenfolge für die unterschiedlichen Ausgabetypen eingehalten.

Standardmäßige Fehlerdatei

Gibt die Datei an, um vom Programm in STDERR geschriebenen Text zu erfassen.

Beispiel:

```
/tmp/trace.err
```

Sie können sowohl für die standardmäßige Fehler- als auch die standardmäßige Ausgabedatei dieselbe Datei angeben. Es wird allerdings keine Reihenfolge für die unterschiedlichen Ausgabetypen eingehalten.

Ausgabe in Protokolle eintragen

Protokolle verarbeiten Ausgaben in die globalen Protokolldateien.

Ausgabe in Datensatzvariable eintragen

Kopiert die Ausgabe eines Betriebsablaufs (stdout und stderr) in die Operatordatensatzvariable processOutput.

Für Standardausgabe verwendete Protokolldatei abschneiden

Ersetzt bei jedem Schreiben von neuer Ausgabe eine vorhandene Protokolldatei mit demselben Namen. Deaktivieren Sie dieses Kontrollkästchen, wenn Ausgabe an eine vorhandene Fehlerprotokolldatei mit demselben Namen angefügt werden soll.

Dieses Kontrollkästchen ersetzt eine vorhandene Datei selbst dann, wenn diese auch für Standardfehler verwendet wird und das Kontrollkästchen "Für Standardfehler verwendete Protokolldatei abschneiden" nicht aktiviert ist.

Für Standardfehler verwendete Protokolldatei abschneiden

Ersetzt bei jedem Schreiben von neuer Ausgabe eine vorhandene Fehlerdatei mit demselben Namen. Deaktivieren Sie dieses Kontrollkästchen, wenn Ausgabe an eine vorhandene Fehlerdatei mit demselben Namen angefügt werden soll.

Dieses Kontrollkästchen ersetzt eine vorhandene Datei selbst dann, wenn diese auch für die Standardausgabe verwendet wird und das Kontrollkästchen "Für Standardfehler verwendete Protokolldatei abschneiden" nicht aktiviert ist.

BS-Anwenderprofil laden

Lädt das Betriebssystemprofil (normalerweise Windows), das dem Anwenderkonto zugeordnet ist. Die Anwender-ID gibt das Profil an (zusätzlich zu dem durch "Profil" angegebenen Profil, das Umgebungsvariablen angibt). Normalerweise wird dieses Profil nicht verwendet, es sei denn, es sollen Zuordnungen sowie ähnliche in der Windows-Registrierung verwaltete Konstrukte für einen bestimmten Anwender eingerichtet werden. Das Herunterladen von Anwenderinformationen von einem Domänenserver führt zu einer Leistungseinbuße.

Prozess am Ende des Ablaufs beenden

Der Betriebssystemprozess, der das bestimmte Programm ausführt, wird beendet (sofern er nicht bereits beendet wurde), sobald der CA Process Automation-Prozess abgeschlossen ist.

Ausgabeparameter

programName

Profil

workingDir

userID

password

parameters

stdOutFileName

stdErrFileName

isPostToGlobalLog

isPostToOutVar

isTruncateForStdOut

isTruncateForStdErr

isLoadOSProfile

isKillProcessOnFlowEnd

processOutput

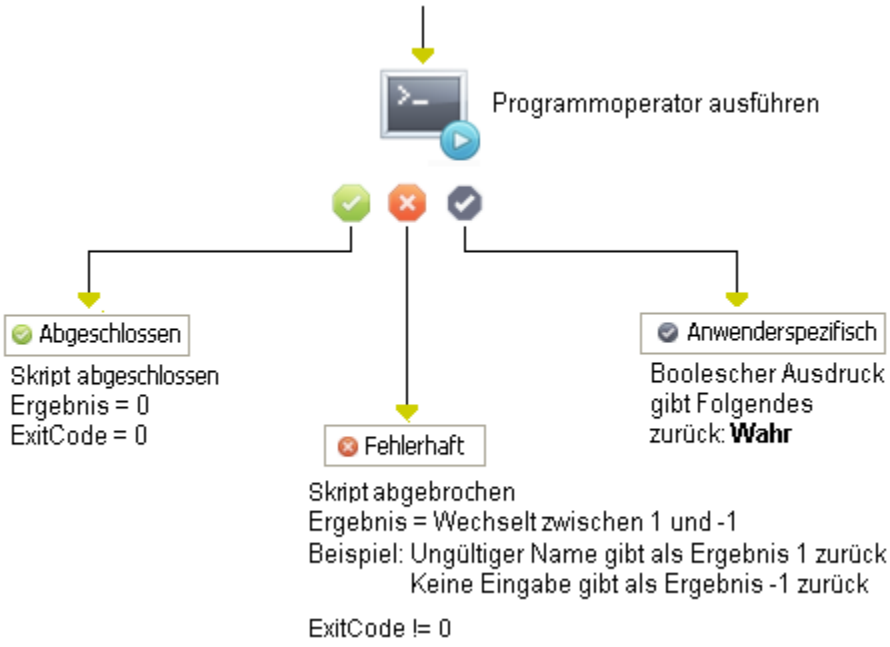
ExitCode

PID

Warnungen

Operator "Skript ausführen"

Der Operator "Skript ausführen" führt ein Skript auf einem Kontaktpunkt-Hostcomputer aus.



Hinweis: Für fast alle Operatoren wird "Ergebnis" bei erfolgreicher Ausführung auf "1" und bei einer fehlerhaften Ausführung auf eine andere Zahl, die nicht "1" entspricht, gesetzt. Für diesen bestimmten Operator wird "Ergebnis" bei erfolgreicher Ausführung auf "0" festgelegt und auf eine andere Zahl, die nicht "0" entspricht, bei einer fehlerhaften Ausführung.

Eingabeparameter

Skript-Erweiterung

Gibt für Windows die Erweiterung an, die den Typ des Skripts kennzeichnet. Wählen Sie eine Option aus der Drop-down-Liste aus oder geben Sie eine Erweiterung ein.

Inline-Skript

Gibt ein Unterskript für das auszuführende Skript an. Klicken Sie auf die Schaltfläche "Durchsuchen (...)" zum Öffnen des Editors, um das auszuführende Skript einzugeben.

UNIX

Das Skript muss ein Skript sein, das den UNIX-Protokollen entsprechend ausgeführt werden kann. Die erste Zeile des Skripts muss den vollständigen Pfad der Shell angeben, die verwendet wird, um das Skript zu interpretieren (beispielsweise #!/bin/sh).

Windows

Das Skript muss ein Skript sein, das entsprechend der im Feld "Skript-Erweiterung" angegebenen Windows-Erweiterung ausgeführt werden kann.

Hinweis: Weitere Informationen über die Verwendung des CA Process Automation Code-Editors finden Sie im *Handbuch für Inhaltsdesign*.

Profil

UNIX

Gibt ein Hostsystemprofil an, das als Quelle verwendet wird, um Teile des Verarbeitungskontextes für das Skript zu definieren (zum Beispiel /home/username/appli_1_profile).

Wenn Sie kein Profil angeben, verwendet das Produkt das in den Eigenschaftseinstellungen der Kategorie "Befehlsausführung" angegebene Profil. Das Produkt verwendet die Anwenderdatei, die den Prozess ausführt, wenn die Eigenschaftseinstellungen der Kategorie "Befehlsausführung" kein Profil angeben.

Windows

Gibt den Pfad zu einer Batch-Datei (.bat) in Windows an, in der Umgebungsvariablen festgelegt werden, die im Prozess verwendet werden sollen. Die Definitionen der Umgebungsvariablen in der Batch-Datei haben folgendes Format:

```
SET SOME_ENV_VAR=/tmp/PAM.exe
SET ANOTHER_ENV_VAR=/tmp/aaaa
```

Die Variableneinstellungen, die in der hier angegebenen Batch-Datei definiert sind, gibt es zusätzlich zu allen Einstellungen, die für das Anwenderprofil definiert sind, mit dem der Prozess ausgeführt wird. Wenn diese Option keine Datei angibt, verwendet der Operator die Einstellungen, die der Wert des Shell-Profilparameters der Kategorie "Befehlsausführung" definiert.

Arbeitsverzeichnis

UNIX

Gibt das Arbeitsverzeichnis für den Operator an (zum Beispiel, /home/user1). Normalerweise ist das Arbeitsverzeichnis der Ordner, der die Programmdatei oder zugehörige Dateien enthält, die das Programm benötigt. Das Produkt sucht in diesem Verzeichnis nach Dateien, die ohne konkrete Pfade angegeben werden.

Wenn Sie kein Arbeitsverzeichnis angeben, wird standardmäßig das Startverzeichnis des Anwenderkontos verwendet, unter dem das Skript ausgeführt wird.

Windows

Gibt das Arbeitsverzeichnis für den Operator an. Normalerweise ist das Arbeitsverzeichnis der Ordner, der die Skriptdatei oder zugehörige Dateien enthält, die das Skript benötigt.

Wenn Sie kein Arbeitsverzeichnis angeben, wird standardmäßig das Arbeitsverzeichnis des Anwenderkontos verwendet, unter dem das Skript ausgeführt wird.

Anwender-ID

Gibt den Anwendernamen an, unter dem das Skript ausgeführt werden soll. Der Ausdruck muss Ausführungsberechtigungen auf der Datei haben (zum Beispiel Process.Appli_1.User). Wenn Sie das Feld "Anwender-ID" leer lassen, wird standardmäßig der Anwender angegeben, den die Konfiguration der Kategorie "Befehlsausführung" angibt.

Normalerweise werden Anwendernamen (und zugehörige Kennwörter) in benannten Datensatzvariablen gespeichert, sodass sie zentral aktualisiert werden können, ohne dass Prozesswerte geändert werden müssen.

Kennwort

Gibt das aktuelle Kennwort an, das mit der angegebenen Anwender-ID verbunden ist. Zum Beispiel wird im folgenden Ausdruck das Kennwort auf den Wert der Prozessvariablen "Password" festgelegt:
`Process.Password`

Das Produkt prüft normalerweise das Kennwort in Hinblick auf die Systeminformationen. Allerdings können Administratoren für Fälle, in denen nicht standardmäßige Sicherheitsmechanismen auf dem Zielhost definiert sind, diese Überprüfung deaktivieren.

Da Sie das Kennwort als Ausdruck angeben, muss der Text, den Sie eingeben, sichtbar sein. Vermeiden Sie daher Literalzeichenfolgen, und verweisen Sie stattdessen auf Datensatzvariablen für Kennwörter.

Parameter

Legt Parameter für das Programm fest, wie im folgenden Beispiel:

```
/tmp/input_file
/tmp/output_file
```

Die Parameter werden in der Reihenfolge an den Prozess übergeben, in der sie aufgelistet sind.

Die Programmparameter werden bei Inbetriebnahme einzeln an das Programm übergeben (das heißt, sie werden nicht mit Leerzeichen miteinander verbunden). Zum Beispiel gibt die Eingabe des folgenden Ausdrucks in einer Zeile den einzelnen Parameter “P1P2” zurück:

```
P1 + P2
```

Wenn folgende Ausdrücke in zwei Zeilen eingegeben werden, werden zwei Parameter zurückgegeben: "P1" und "P2":

```
"P1"
"P2"
```

Standardmäßige Ausgangsdatei

Zeigt die Standardausgabendatei für das Skript an. Wenn Sie nicht den vollständigen Pfad angeben, definiert der Parameterwert des Arbeitsverzeichnisses das Stammverzeichnis für den Pfad (zum Beispiel /tmp/trace.log).

Die Befehlsausführungskategorie leitet den stdout-Datenstrom aus dem Prozess in die angegebene Datei. Sie können sowohl für die standardmäßige Fehler- als auch die standardmäßige Ausgabedatei dieselbe Datei angeben. Allerdings hält sich das Produkt für die unterschiedlichen Ausgabearten an keine relative Reihenfolge.

Standardmäßige Fehlerdatei

Zeigt die Standardfehlerdatei für das Skript an. Wenn Sie nicht den vollständigen Pfad angeben, definiert der Parameterwert des Arbeitsverzeichnisses das Stammverzeichnis für den Pfad (zum Beispiel /tmp/trace.err).

Die Befehlsausführungskategorie leitet den stderr-Datenstrom aus dem Prozess in die angegebene Datei. Sie können sowohl für die standardmäßige Fehler- als auch die standardmäßige Ausgabedatei dieselbe Datei angeben. Allerdings hält sich das Produkt für die unterschiedlichen Ausgabearten an keine relative Reihenfolge.

Ausgabe in Protokolle eintragen

Protokolle verarbeiten Ausgaben in die globalen Protokolldateien.

Ausgabe in Datensatzvariable eintragen

Kopiert die Ausgabe eines Betriebsablaufs (stdout und stderr) in die Variable "Operatordatensatz" (zum Beispiel "scriptOutput").

Für Standardausgabe verwendete Protokolldatei abschneiden

Aktivieren Sie dieses Kontrollkästchen, damit das Produkt eine vorhandene Protokolldatei ersetzt, die den selben Namen hat, wenn sie eine neue Ausgabe schreibt.

Wenn das Kontrollkästchen aktiviert ist, ersetzt das Produkt eine vorhandene Datei, selbst wenn die folgenden Elemente zutreffen:

- Die Datei wird auch für standardmäßige Fehlerausgabe verwendet.
- Das Kontrollkästchen "Für Standardfehler verwendete Protokolldatei abschneiden" ist deaktiviert.

Wenn das Kontrollkästchen deaktiviert ist, hängt das Produkt die Ausgabe an eine vorhandene Fehlerprotokolldatei mit dem selben Namen an.

Für Standardfehler verwendete Protokolldatei abschneiden

Aktivieren Sie dieses Kontrollkästchen, damit das Produkt eine vorhandene Fehlerdatei ersetzt, die den selben Namen hat, wenn sie eine neue Ausgabe schreibt.

Wenn das Kontrollkästchen aktiviert ist, ersetzt das Produkt eine vorhandene Datei, selbst wenn die folgenden Elemente zutreffen:

- Die Datei wird auch für standardmäßige Ausgabe verwendet.
- Das Kontrollkästchen "Für Standardausgabe verwendete Protokolldatei abschneiden" ist deaktiviert.

Wenn das Kontrollkästchen deaktiviert ist, hängt das Produkt die Ausgabe an eine vorhandene Fehlerdatei mit dem selben Namen an.

BS-Anwenderprofil laden

Lädt das Betriebssystemprofil (normalerweise Windows), das den folgenden Elementen zugeordnet ist.

- Das Anwenderkonto, das die Anwender-ID angibt.
- Das Profil, das "Profil" festlegt, das Umgebungsvariablen definiert

Das BS-Anwenderprofil wird normalerweise nur verwendet, um Zuordnungen und ähnliche auf Windows-Registrierung basierte Einrichtungen für einen bestimmten Anwender zu erstellen. Das Herunterladen von Anwenderinformationen von einem Domänenserver führt zu einer Leistungseinbuße.

Prozess am Ende des Ablaufs beenden

Wenn Sie diese Option auswählen, beendet das Produkt den Prozess, wenn der Prozessablauf abgeschlossen ist.

PowerShell-Ausführungsrichtlinie

Um PowerShell-Skripte auszuführen, legt Windows eine Sicherheit bezüglich der Ausführungsrichtlinie fest. Die Windows PowerShell-Ausführungsrichtlinie bestimmt, ob Skripte ausgeführt werden dürfen, und wenn ja, ob sie digital signiert werden müssen. Sie bestimmt außerdem, ob Konfigurationsdateien geladen werden können.

Die standardmäßige Ausführungsrichtlinie von PowerShell unter Windows ist beschränkt. Um ein PowerShell-Skript auszuführen, ändern Sie die Ausführungsrichtlinie in einen der folgenden Werte:

- RemoteSigned
- AllSigned
- Unrestricted

CA Process Automation gibt während der Installation des Agenten oder Koordinationsrechners eine Option an, um die Ausführungsrichtlinie des PowerShell-Skripts auf "Remote Signed" (per Remote-Zugriff signiert) festzulegen (d. h., dass heruntergeladene Skripte von einem vertrauenswürdigen Herausgeber signiert werden müssen, bevor sie ausgeführt werden können). Allerdings haben Sie immer die Möglichkeit, die Ausführungsrichtlinie über die Eingabeaufforderung zu ändern, indem Sie folgenden PowerShell-Befehl verwenden:

```
Set-ExecutionPolicy
```

...gefolgt vom entsprechenden Richtliniennamen. Zum Beispiel legt dieser Befehl die Ausführungsrichtlinie auf "AllSigned" fest:

```
Set-ExecutionPolicy AllSigned
```

Ausgabeparameter

scriptType

inLineScript

Profil

workingDir

userID

password

parameters

stdOutFileName

stdErrFileName

isPostToGlobalLog

isPostToOutVar

isTruncateForStdOut

isTruncateForStdErr

isLoadOSProfile

isKillProcessOnFlowEnd

processOutput

StartDate

StartTime

Ergebnis

ExitCode

PID

Ursache

Warnungen

Operator "SSH-Befehl ausführen"

Der Operator "SSH-Befehl ausführen" wurde für die Verwendung mit Zielen, beispielsweise für Netzwerkgeräte oder andere Nicht-Servergeräte, entworfen

Für die Ausführung auf Remote-Servern mit SSH kann es einfacher sein, die Proxy-Kontaktpunkt- oder die Hostgruppen-Konzepte zu verwenden.

Hinweis: Dieser Operator erfordert nicht, dass der Anwender die Anmeldungssequenz angibt.

Der Operator "SSH-Befehl ausführen" führt folgende Aktionen aus:

- Stellt eine SSH-Verbindung zum Remote-Host her.
- Sendet gleichzeitig einen Befehl.
- Liest die Ausgabe des Befehls, bis die Eingabeaufforderung angezeigt wird, um anzuzeigen, dass der Befehl abgeschlossen ist.
- Sendet den nächsten Befehl.

Sie können die Zeit festlegen, die maximal auf die Eingabeaufforderung gewartet werden soll, bevor der Operator fehlschlägt. Stellen Sie sicher, dass diese Einstellung größer ist als die Ausführungszeit des längsten Befehls, den dieser Operator ausführen kann.

Sie können festlegen, dass dieser Operator nach Anmeldung und bevor der Befehl ausgeführt wird, zu einem anderen Anwender (einschließlich "Root") wechselt. Das Wechseln von Anwendern ermöglicht es, dass die Befehle unter einem anderen Anwender ausgeführt werden. Das Wechseln zu einem anderen Anwender wird interaktiv durchgeführt.

Eingabeparameter

Für alle Eingaben, die in diesem Operator als regulärer Ausdruck angegeben werden können, stimmt der Operator die gesamten Antwortdaten mit dem Muster ab. Der Operator "SSH-Befehl ausführen" entspricht nicht dem Muster als untergeordnete Zeichenfolge der Antwortdaten. Ein Punkt ‘.’ entspricht einem neuen Zeilenabschluss (kann verwendet werden, um mehrzeilige Antwortdaten abzustimmen).

Befehle

Name des Remote-Host

Der Hostname oder die IP des Computers, zu der eine Verbindung hergestellt werden soll.

Indizierte Zeichenfolgenvariable für Befehle verwenden?

Wenn dieses Kontrollkästchen nicht aktiviert ist, können Sie Befehle in das Feld "Befehle" eingeben. Wählen Sie dieses Feld aus, um die Befehle als indizierte Zeichenfolgenvariablen im Feld "Befehle - Indizierte Zeichenfolgenvariable" anzugeben.

Befehle

Liste der Befehle, die auf dem Remote-Host ausgeführt werden sollen. Beenden Sie die Liste nicht mit dem Befehl "exit", da der Operator die SSH-Sitzung automatisch verlässt, nachdem der letzte Befehl ausgeführt wurde.

Befehle - Indizierte Zeichenfolgenvariable

Der Name der Datensatzvariable, die eine Liste von Befehlen enthält, die auf dem Remote-Host ausgeführt werden sollen. Beenden Sie die Liste nicht mit dem Befehl "exit", da der Operator die SSH-Sitzung automatisch verlässt, nachdem der letzte Befehl ausgeführt wurde.

Ausgabe in Datensatzvariable speichern?

Aktivieren Sie dieses Kontrollkästchen, um die Ausgabe aller Befehle in den Datensatz des Operators zu kopieren. Die Ausgabe von jedem Befehl ist in der Variable "SSHCommandsOutput" gespeichert.

Größenbeschränkung für die Befehlsausgabe

Gibt die Höchstanzahl von Byte jeder Befehlsausgabe an, die jeweils in der Datensatzvariable des Operators gespeichert werden soll. Wenn diese Zahl nicht angegeben ist, verwendet der Operator den Wert: 4096.

Eingabeaufforderung für Anwender

Dieses Feld erfüllt zwei Zwecke:

- Es zeigt an, dass der Anwender angemeldet ist. Der Operator sucht nach dieser Eingabeaufforderung, nachdem der Anwender sich angemeldet hat.
- Zeigt an, dass ein Befehl (in der Liste der Befehle oder eines Befehls-Array, der in der SSH-Sitzung unter diesem Anwender ausgeführt wurde) beendet wurde. Der Operator kann dann den nächsten Befehl in der Liste senden.

Dieses Feld ist im Allgemeinen eine Anzeige der Eingabeaufforderung des Anwenders. Das Feld wird normalerweise als "#", "$, ">" usw. angegeben. Allerdings muss es als regulärer Ausdruck angegeben werden. Zum Beispiel: ".*[$>?:#]", um alle Eingaben (einschließlich neuer Zeilen) zu finden, auf die "$", ">", "?", ":" oder "#" folgt. Geben Sie alle Aufforderungen an, die während der Ausführung der Befehle angezeigt werden sollen.

Wichtig! Starten Sie den regulären Ausdruck mit .*, um alle Daten zu finden, die durch den Befehl zurückgegeben wurden, bis die Aufforderung angezeigt wird. Dieser reguläre Ausdruck sollte bis zur nächsten Aufforderung alle Ausgaben des Befehls finden.

Hinweis: Die Klammern um "$" sind erforderlich, um das $-Zeichen anzuzeigen. "$" hat eine in regulären Ausdrücken eine besondere Bedeutung, wenn es nicht in Klammern gesetzt wird.

Zeit für Warten auf Aufforderungen (Sek.):

Die Zeit (in Sekunden), während der auf eine Aufforderung zum Senden von Befehlen gewartet wird. Wenn dieses Feld leer gelassen wird, verwendet der Operator den Wert: 60.

Dieses Feld gilt für Aufforderungen, die erwartet werden, nachdem alle im Operator angegebenen Befehle ausgeführt wurden. Der Operator weiß nicht, ob ein in der SSH-Sitzung ausgeführter Befehl alle Daten zurückgegeben hat. Der Operator fährt mit dem Lesen der Ausgabe des Befehls fort, bis eine Übereinstimmung mit dem angegebenen Anwender oder mit einer Eingabeaufforderung für Anwenderwechsel vorhanden ist oder bis dieses Zeitlimit abgelaufen ist (je nachdem, was zuerst zutrifft). Der Operator fährt dann mit der Verarbeitung der Ausgabe des Befehls fort, bevor zum nächsten Befehl gewechselt wird oder der Operator fehlschlägt.

Wichtig! Stellen Sie sicher, dass diese Einstellung größer ist als die Ausführungszeit des längsten Befehls, den dieser Operator ausführen kann.

Informationen für Remote-Anmeldung

Pseudoterminaltyp

Der Pseudoterminaltyp, der bei SSH-Verbindungen angefordert wird. Dieses Feld überschreibt den Wert, der auf der Kategorieebene angegeben ist. Wenn das Feld leer gelassen wird, verwendet der Operator den Standardwert, der auf der Kategorieebene festgelegt ist. Wenn dieser Wert leer ist, wird der Operator standardmäßig auf VT100 festgelegt.

- VT100 funktioniert normalerweise mit den meisten Computern (besonders Linux-basiert).
- VT400 funktioniert normalerweise mit Windows-basierten Computern. VT400 ist für Windows-Plattformen erforderlich, besonders wenn die Ausgabe, die aus dem SSH-Server (Befehlsausgabe) abgerufen wird, Steuerzeichen anstatt Leerzeichen enthält. Zum Beispiel wenn in der Ausgabe "[19;1H" statt Leerzeichen enthalten ist. VT400 interpretiert die Leerzeichen für Windows richtig.

Andere Terminaltypen können verwendet werden. Stellen Sie sicher, dass Sie sie testen, bevor Sie den Operator in die Produktion verschieben. Einige Pseudoterminaltypen sind:

- dumb
- xterm
- vt220
- vt320
- gogrid

Überprüfen Sie die Installation und Konfiguration Ihres SSH-Servers für den unterstützten Pseudoterminal. Einige SSH-Server listen die unterstützten Pseudoterminal im Ordner "TermInfo" auf.

Der Pseudoterminaltyp steuert, wie Leerzeichen in der Befehlsausgabe angezeigt werden. Sie sollten diesen Operator gegen die Pseudoterminals testen, die vom SSH-Server unterstützt werden, um einen geeigneten Pseudoterminal zu finden, der die Leerzeichen entsprechend zurückgibt. Wenn die Leerzeichen nicht entsprechend zurückgegeben werden und kein Pseudoterminal, der diesen Issue beheben könnte, verfügbar ist, gehen Sie folgendermaßen vor:

- Ändern Sie die Eingabe des Operators, um die Beschränkung dieses SSH-Servers unterzubringen.
- Verwenden Sie JavaScript, um die Ausgabe der Befehle zu verfeinern bzw. zu extrahieren.

Wenn Sie einen Pseudoterminal anfordern, der nicht unterstützt wird, geben einige SSH-Server einen Fehler zurück, während andere Server den angeforderten Pseudoterminaltyp ignorieren und andere Pseudoterminaltypen verwenden. Überprüfen Sie die Protokolle des SSH-Servers für den Pseudoterminal, die verwendet werden, wenn der Operator ausgeführt wird.

Port

Der Port zum Verbinden mit dem Remote-Host. Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist. Wenn dieses Feld leer gelassen wird, verwendet der Operator den Standardwert, der auf der Operatorkategorieebene festgelegt ist. Wenn dieser Standardoperatorkategorienwert leer ist, verwendet der Operator den Wert 22.

Anwendername:

Der Anwendername, der für die Anmeldung beim Remote-Host verwendet wird. Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist. Wenn dieses Feld leer gelassen wird, verwendet der Operator den Standardwert, der auf der Operatorkategorieebene festgelegt ist.

Privaten Schlüssel für die Anmeldung verwenden?

Gibt an, ob ein privater Schlüssel für die Anmeldung am Remote-Host verwendet werden soll (statt Kennwortinformationen). Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist. Wenn dieses Feld leer gelassen wird, verwendet der Operator den Standardwert, der auf der Operatorkategorieebene festgelegt ist. Wählen Sie eine der folgenden Optionen aus:

Wahr

Fordert den Operator auf, einen privaten Schlüssel zu verwenden.

Wenn "Wahr" ausgewählt ist, werden folgende Felder aktiviert: "Eingabequelle des privaten Schlüssels", "Inline-Inhalt des privaten Schlüssels", "Privater Schlüssel als Ausdruck", "Pfad des privaten Schlüssels", "Passphrase für Schlüssel". Das Feld "Kennwort" wird deaktiviert.

False

Fordert den Operator auf, Kennwortinformationen zu verwenden.

Wenn "Falsch" ausgewählt ist, werden folgende Felder deaktiviert: "Eingabequelle des privaten Schlüssels", "Inline-Inhalt des privaten Schlüssels", "Privater Schlüssel als Ausdruck", "Pfad des privaten Schlüssels", "Passphrase für Schlüssel". Das Feld "Kennwort" wird aktiviert.

Wenn andere Werte eingegeben werden, wird der Operator aufgefordert, den Wert "Falsch" zu verwenden und aktiviert alle Felder (um den Anwender unterzubringen, der einen Ausdruck eingibt).

Kennwort

Das Kennwort, das für die Anmeldung beim Remote-Host verwendet wird. Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist. Wenn dieses Feld leer gelassen wird, verwendet der Operator den Standardwert, der auf der Operatorkategorieebene festgelegt ist.

Eingabequelle des privaten Schlüssels

Gibt an, wie der private Schlüssel angegeben werden soll. Wählen Sie eine der folgenden Optionen aus:

Inline-Inhalt

Aktiviert "Inline-Inhalt des privaten Schlüssels" und deaktiviert "Privater Schlüssel als Ausdruck" und "Pfad des privaten Schlüssels".

Dateipfad

Aktiviert "Pfad des privaten Schlüssels" und deaktiviert "Inline-Inhalt des privaten Schlüssels" und "Privater Schlüssel als Ausdruck".

Ausdruck

Aktiviert "Privater Schlüssel als Ausdruck" und deaktiviert "Inline-Inhalt des privaten Schlüssels" und "Pfad des privaten Schlüssels".

Inline-Inhalt des privaten Schlüssels

Der Inhalt des privaten Schlüssels für die Anmeldung beim Remote-Host. Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist. Wenn dieses Feld leer gelassen wird und "Eingabequelle des privaten Schlüssels" auf "Inline-Inhalt" festgelegt wird, verwendet der Operator den Standardwert, der auf der Operatorkategorieebene festgelegt ist.

Pfad des privaten Schlüssels

Der Pfad zum privaten Schlüssel für die Anmeldung beim Remote-Host. Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist. Wenn dieses Feld leer gelassen wird und "Eingabequelle des privaten Schlüssels" auf "Dateipfad" festgelegt wird, verwendet der Operator den Standardwert, der auf der Operatorkategorieebene festgelegt ist.

Privater Schlüssel als Ausdruck

Die Datensatzvariable, die den Inhalt des privaten Schlüssels für die Protokollierung im Remote-Host enthält.

Hinweis: Stellen Sie sicher, dass die Datensatzvariable eine mehrzeilige Zeichenfolge ist.

Passphrase für Schlüssel

Optionale Passphrase, um den Inhalt des privaten Schlüssels zu entsperren. Dieses Feld ist erforderlich, wenn der private Schlüssel mit einer Passphrase erstellt wurde. Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist.

Wenn "Passphrase für Schlüssel" leer ist, wird der Operator nicht automatisch aufgefordert, den Wert "Passphrase für Schlüssel" aus den Operatorkategorieeinstellungen zu erben. Tatsächlich wird das Feld "Passphrase für Schlüssel" folgendermaßen mit dem Feld "Inline-Inhalt des privaten Schlüssels", "Pfad des privaten Schlüssels" oder "Privater Schlüssel als Ausdruck" verbunden:

- Wenn die "Passphrase für Schlüssel" des Operators angegeben ist, wird sie durch den Operator verwendet.
- Wenn die "Passphrase für Schlüssel" des Operators leer ist, wird "Inline-Inhalt des privaten Schlüssels" angegeben (nicht leer) und der Wert von "Eingabequelle des privaten Schlüssels" wird auf "Inline-Inhalt" festgelegt. Dann verwendet der Operator eine leere Passphrase für Schlüssel (Passphrase nicht festgelegt).

- Wenn die "Passphrase für Schlüssel" des Operators leer ist, wird "Pfad des privaten Schlüssels" angegeben (nicht leer) und der Wert von "Eingabequelle des privaten Schlüssels" wird auf "Dateipfad" festgelegt. Dann verwendet der Operator eine leere Passphrase für Schlüssel (Passphrase nicht festgelegt).
- Wenn die "Passphrase für Schlüssel" des Operators leer ist, wird "Privater Schlüssel als Ausdruck" angegeben (nicht leer) und der Wert von "Eingabequelle des privaten Schlüssels" wird auf "Ausdruck" festgelegt. Dann verwendet der Operator eine leere Passphrase für Schlüssel (Passphrase nicht festgelegt).
- Für alle anderen Fälle verwendet der Operator "SSH-Befehl ausführen" die Option "Standardmäßige Passphrase für Schlüssel".

Hinweis: Die Erstellung von privaten bzw. öffentlichen Schlüsseln für SSH wird im *Administrationshandbuch* beschrieben.

Kennwort zu Anwenderwechsel

Befehl bzw. Skript als anderer Anwender ausführen?

Sollte die angegebenen Befehle als ein anderer Anwender ausgeführt werden? Wählen Sie "Wahr", um Anwender nach Anmeldung zu wechseln oder "Falsch", um den Vorgang weiterhin als Anmeldungsanwender auszuführen.

Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist. Wenn dieses Feld leer gelassen wird, verwendet der Operator den Standardwert, der auf der Operatorkategorieebene festgelegt ist.

Befehl für Anwenderwechsel

Der Befehl, um den Anwender auf dem Remote-Host zu wechseln. Dies ist im Allgemeinen:

- su - Anwendername

  oder

- sudo su - Anwendername

Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist. Wenn dieses Feld leer gelassen wird, verwendet der Operator den Standardwert, der auf der Operatorkategorieebene festgelegt ist.

Textaufforderung für Kennwort bei Anwenderwechsel

Die Textaufforderung, die anzeigt, dass der Remote-Host für das Wechseln zwischen verschiedenen Anwendern ein Kennwort benötigt. Dies ist im Allgemeinen:

- Kennwort:

  oder

- kennwort:

Dieser Parameter muss als regulärer Ausdruck angegeben werden. Zum Beispiel, ".*ennwort: ", um alle Eingaben zu finden, auf die "ennwort: " folgt (einschließlich neuer Zeilen).

Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist. Wenn dieses Feld leer gelassen wird, verwendet der Operator den Standardwert, der auf der Operatorkategorieebene festgelegt ist.

Wenn für das Wechseln zwischen verschiedenen Anwendern kein Kennwort erforderlich ist (zum Beispiel, wenn man von Root zu einem anderen Anwender gewechselt wird), können Sie einen beliebigen Wert in dieses Feld eingeben. Der Operator versucht, die Daten zu finden, die aus der SSH-Sitzung gelesen wurden, nachdem zunächst "Eingabeaufforderung für Anwenderwechsel" gegen "Textaufforderung für Kennwort bei Anwenderwechsel" gesendet wurde. Wenn keine Übereinstimmung vorhanden ist, wird versucht, die Daten mit der Eingabeaufforderung für Anwenderwechsel abzustimmen, um zu überprüfen, ob ein Kennwort erforderlich ist.

Kennwort für Anwenderwechsel

Das Kennwort zum Wechseln von einem Anwender zu einem anderen. Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist. Wenn dieses Feld leer gelassen wird, verwendet der Operator den Standardwert, der auf der Operatorkategorieebene festgelegt ist.

Dieses Feld wird nicht verwendet, wenn für das Wechseln zu einem anderen Anwender kein Kennwort erforderlich ist.

Eingabeaufforderung für Anwenderwechsel

Dieses Feld erfüllt zwei Zwecke:

- Um anzuzeigen, dass der Anwenderwechsel stattgefunden hat.
- Um anzuzeigen, dass ein Befehl in der Liste der Befehle oder des Befehls-Arrays, der in der SSH-Sitzung unter dem neuen Anwender (zu dem der Operator wechselt) ausgeführt wurde, beendet wurde und dass der Operator den nächsten Befehl in der Liste senden kann.

Dieses Feld ist im Allgemeinen eine Anzeige der Eingabeaufforderung des neuen Anwenders (zu dem der Operator wechselt). Es ist im Allgemeinen "#", $, ">" usw. Dieses Feld muss als regulärer Ausdruck angegeben werden. Zum Beispiel: ".*[$>?:#]", um alle Eingaben (einschließlich neuer Zeilen) zu finden, auf die "$", ">", "?", ":" oder "#" folgt. Geben Sie alle Aufforderungen an, die während der Ausführung der Befehle angezeigt werden sollen. Starten Sie den regulären Ausdruck mit .*, um alle Daten zu finden, die durch den Befehl zurückgegeben wurden, bis die Aufforderung angezeigt wird. Dieser reguläre Ausdruck findet bis zur nächsten Aufforderung alle Ausgabe des Befehls.

Hinweis: Die Klammern um "$" sind erforderlich, um das $-Zeichen anzuzeigen. "$" hat eine in regulären Ausdrücken eine besondere Bedeutung, wenn es nicht in Klammern gesetzt wird.

Seien Sie vorsichtig mit den RegEx, um falsche Positive zu vermeiden, wie z. B.:

Der Anwender gibt ein ungültiges Kennwort ein, wenn er zum Root wechselt:

```
# su – root
Kennwort:
```

Die Antwort für ein ungültiges Kennwort endet mit #:

```
su: Sorry
#
```

Gibt die gleiche Aufforderung, wie wenn der Anwender ein gültiges Kennwort eingibt, wo die Antwort auch mit # endet:

```
Sun Microsystems Inc.   SunOS 5.10      Generic January 2005
#
```

Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist. Wenn dieses Feld leer gelassen wird, verwendet der Operator den Standardwert, der auf der Operatorkategorieebene festgelegt ist.

Ausgabeparameter

SSHSwitchUserPasswordPrompt

Die eigentliche Textaufforderung für Kennwort bei Anwenderwechsel liest aus der SSH-Sitzung innerhalb der "Zeit für Warten auf Aufforderungen". Dies sind die Daten, die mit den Mustern abgeglichen werden, die im Feld "Textaufforderung für Kennwort bei Anwenderwechsel" angegeben sind. Dieses Feld ist leer, wenn zum Wechseln zwischen Anwendern kein Kennwort erforderlich ist, wie z. B. wenn von Root zu einem anderen Anwender gewechselt wird oder wenn der Operator nicht zum Wechseln zwischen Anwendern eingerichtet ist.

SSHSwitchUserCommandPrompt

Die tatsächliche Eingabeaufforderung für Anwenderwechsel liest das erste Mal aus der SSH-Sitzung (innerhalb "Zeit für Warten auf Aufforderungen"), nachdem zum neuen Anwender gewechselt wurde. Dies sind die Daten, die mit den Mustern abgeglichen werden, die im Feld "Eingabeaufforderung für Anwenderwechsel" angegeben sind. Dieses Feld ist leer, wenn der Operator nicht zum Wechseln zwischen Anwendern eingerichtet ist.

SSHCommandsOutput

Eine indizierte Zeichenfolge mit der Ausgabe, die aus jedem Befehl (innerhalb der "Zeit für Warten auf Aufforderungen") gelesen wurde, der auf dem Remote-Host ausführt wurde. Die Ausgabe für jeden Befehl ist auf den Wert gekürzt, der in der Größenbeschränkung für die Befehlsausgabe angegeben ist.

Die vollständige Ausgabe von jedem Befehl enthält die nächste Aufforderung.

SSHHost

CommandsAsAnArray

parameters

CommandsArray

isPostToOutVar

PostToOutVarSizeLimit

SSHHostUserCommandPrompt

SSHHostWaitTimeout

SSHPseudoTerminalType

SSHPort

SSHUsername

SSHUsePrivateKey

SSHPassword

SSHPrivateKeyOption

SSHPrivateKey

SSHPrivateKeyVar

SSHPrivateKeyPath

SSHPrivateKeyPassphrase

SSHSwitchUser

SSHHostSwitchUserCommand

SSHHostSwitchUserPasswordPrompt

SSHHostSwitchUserPassword

SSHHostSwitchUserCommandPrompt

Operator-Ports

Erfolg

- Der Operator wurde erfolgreich abgeschlossen.
- Der Operator hängt von den Mustern ab, die in "Eingabeaufforderung für Anwender" und "Eingabeaufforderung für Anwenderwechsel" angegeben sind, um zu entscheiden, ob ein Befehl (und der Operator) erfolgreich ist oder fehlschlägt.

  Zum Beispiel führt das Muster ".*#" zu folgendem Befehl: "badCmd"

  1. Der Operator führt den Befehl aus:

     ```
     # badCmd
     ```

  2. Die Ausgabe wird gelesen (die mit der Aufforderung ".*#" übereinstimmt):

     ```
     badCmd:  not found
     #
     ```

  3. Führen Sie den nächsten Befehl aus:

Fehlgeschlagen

- Der Anwender gibt ungültige Eingaben an. Zum Beispiel negatives Aufforderungs-Zeitlimit, leere Eingabeaufforderung für Anwender usw.
- Es kann keine Verbindung zum Remote-Host hergestellt werden. Überprüfen Sie den Remote-Host, den Anwendernamen, das Kennwort und die Schlüssel, die dem Operator bereitgestellt werden.
- Der Anwender gibt einen unbekannten Host im Remote-Host an.
- Der Anwender kann auf dem Remote-Host nicht authentifiziert werden:
  - Anwender-/Kennwortauthentifizierung: Anwendername oder Kennwort ist ungültig
  - Anwender-/private Schlüsselauthentifizierung:

    - Eine ungültige Passphrase wird für einen passphrasengeschützten privaten Schlüssel angegeben: Der Operator zeigt an, dass der Schlüssel aufgrund eines internen E-/A-Problems nicht gelesen werden kann

    - Es fehlt die Passphrase für einen passphrasengeschützten privaten Schlüssel: Der Operator zeigt an, dass die erforderliche Passphrase für Schlüsselauthentifizierung nicht angegeben ist.

    - Ein ungültiger privater Schlüssel wird angegeben: Der Operator zeigt an, dass der private Schlüssel nicht analysiert werden kann, das Schlüsselformat nicht unterstützt wird oder der Schlüssel aufgrund eines internen E-/A-Problems nicht gelesen werden kann.

    - Der Pfad zur privaten Schlüsseldatei ist ungültig, existiert nicht oder kann nicht gelesen werden.

- Es kann nicht zu einem anderen Anwender gewechselt werden. Die Daten, die aus der SSH-Sitzung (innerhalb der "Zeit für Warten auf Aufforderungen") gelesen wurden, konnten nicht mit dem angegebenen Muster "Eingabeaufforderung für Anwenderwechsel" abgestimmt werden.
- Es kann nicht zu einem anderen Anwender gewechselt werden. Die Daten, die aus der SSH-Sitzung (innerhalb der "Zeit für Warten auf Aufforderungen") gelesen wurden, konnten nicht mit dem angegebenen Muster "Textaufforderung für Kennwort bei Anwenderwechsel" abgestimmt werden.
- Der angegebene Anwendername bzw. das angegebene Kennwort sind nicht berechtigt, sich über SSH anzumelden.
- Der Operator konnte die Eingabeaufforderung (in "Eingabeaufforderung für Anwender" oder "Eingabeaufforderung für Anwenderwechsel" angegeben) nicht mit der Ausgabe abstimmen, die innerhalb der "Zeit für Warten auf Aufforderungen" aus einem Befehl gelesen wurde. In diesem Fall entweder:
  - Der Befehl hat die Ausführung fertiggestellt und Daten zurückgegeben, die nicht mit dem Muster in "Eingabeaufforderung für Anwender" oder "Eingabeaufforderung für Anwenderwechsel" übereinstimmen.
  - Die Ausführung des Befehls dauert länger als die "Zeit für Warten auf Aufforderungen", und der Operator hält das Lesen der Ausgabedaten an, bevor der Befehl beendet wird. Dadurch wurden die gelesenen Daten nicht mit dem Muster in "Eingabeaufforderung für Anwender" oder "Eingabeaufforderung für Anwenderwechsel" abgestimmt. In diesem Fall muss der Wert in "Zeit für Warten auf Aufforderungen" größer sein als die Ausführungszeit des längsten Befehls in der Liste.

  Hinweis: In diesem Fall werden nachfolgende Befehle in der Liste nicht ausgeführt, und "SSHCommandsOutput" enthält nur die Ausgabe der Befehle, die vorher ausgeführt wurden (einschließlich der Ausgabe, die für den aktuellen Befehl gelesen wurde).

Anwenderspezifische Ports

Wenn während des Prozessdesigns durch den Anwender festgelegt.

Beispiel

In diesem Beispiel wird die interaktive Funktionsweise des Operators "SSH-Befehl ausführen" beschrieben.

Gehen Sie folgendermaßen vor::

1. Eine SSH-Verbindung zum Remote-Host wird hergestellt.
2. Mithilfe einer der folgenden Optionen erfolgt eine nicht-interaktive Anmeldung:
   - Ein Anwendername und ein Kennwort.
   - Ein Anwendername und ein privater Schlüssel.
   - Ein Anwendername, ein privater Schlüssel und eine Passphrase, wenn der private Schlüssel mit einer Passphrase erstellt wurde.
3. Es wird aus der SSH-Sitzung gelesen, bis eine der folgenden Aktionen eintritt:
   - Es ist eine Übereinstimmung der Muster vorhanden, die in der *Eingabeaufforderung für Anwender* angegebenen sind. Fahren Sie mit Schritt 4 fort.
   - Der Wert in "Zeit für Warten auf Aufforderungen" läuft ab, ohne dass eine Übereinstimmung der Muster vorhanden ist, die in der Eingabeaufforderung für Anwender angegebenen sind. Brechen Sie den Operator ab.
4. Es wird bestimmt, ob der Operator auf *Befehl als anderer Anwender ausführen* festgelegt werden soll.
   - Wenn ja, fahren Sie mit Schritt 5 fort.
   - Wenn nein, fahren Sie mit Schritt 6 fort.
5. Senden Sie den *Befehl für Anwenderwechsel* und lesen Sie aus der SSH-Sitzung, bis eine der folgenden Aktionen eintritt:
   - Es ist eine Übereinstimmung der Muster vorhanden, die in *Textaufforderung für Kennwort bei Anwenderwechsel* angegeben sind, für die ein Kennwort zum Wechseln der Anwender erforderlich ist.
     - Senden Sie das Kennwort für Anwenderwechsel.
     - Es wird aus der SSH-Sitzung gelesen, bis eine der folgenden Aktionen eintritt:

       Es ist eine Übereinstimmung der Muster vorhanden, die in der *Eingabeaufforderung für Anwenderwechsel* angegebenen sind. Fahren Sie mit Schritt 6 fort.

       Der Wert in "Zeit für Warten auf Aufforderungen" läuft ab, ohne dass eine Übereinstimmung der Muster vorhanden ist, die in der Eingabeaufforderung für Anwenderwechsel angegebenen sind. Brechen Sie den Operator ab.

- Es ist eine Übereinstimmung der Muster vorhanden, die in der *Eingabeaufforderung für Anwenderwechsel* angegeben sind, für die kein Kennwort zum Wechseln der Anwender erforderlich ist. Fahren Sie mit Schritt 6 fort.
- Der Wert in "Zeit für Warten auf Aufforderungen" läuft ab, ohne dass eine Übereinstimmung der *Textaufforderung für Kennwort bei Anwenderwechsel* und der *Eingabeaufforderung für Anwenderwechsel* vorhanden ist. Brechen Sie den Operator ab.

6. Der Operator "SSH-Befehl ausführen" hat sich angemeldet und gegebenenfalls den Anwender gewechselt. Der Operator ist nun bereit, die Befehle auszuführen.
7. Führen Sie die Befehle in einer Schleife aus, senden Sie gleichzeitig einen Befehl, und lesen Sie aus der SSH-Sitzung, bis eine der folgenden Aktionen eintritt:
   - Es ist eine Übereinstimmung der Muster vorhanden, die in der *Eingabeaufforderung für Anwender* angegeben sind, wenn nicht zu einem anderen Anwender gewechselt wurde. Wiederholen Sie Schritt 7, und senden Sie den nächsten Befehl.
   - Es ist eine Übereinstimmung der Muster vorhanden, die in der *Eingabeaufforderung für Anwenderwechsel* angegeben sind, wenn zu einem anderen Anwender gewechselt wurde. Wiederholen Sie Schritt 7, und senden Sie den nächsten Befehl.
   - Der Wert in "Zeit für Warten auf Aufforderungen" läuft ab, ohne dass eine "Eingabeaufforderung für Anwender" oder "Eingabeaufforderung für Anwenderwechsel" (je nachdem, was zutreffend ist) vorhanden ist. Brechen Sie den Operator ab.

Operator "SSH-Skript ausführen"

Der Operator "SSH-Skript ausführen" wird im interaktiven Modus ausgeführt, um Netzwerkgeräte unterzubringen, bei denen des Vorhandensein eines Dateisystems unbekannt ist.

Hinweis: Verwenden Sie für die nicht interaktive SSH-Kommunikation einen Proxy-Kontaktpunkt oder eine Hostgruppe.

Der Operator verwendet die von Ihnen angegebenen Anmeldeinformationen, um folgende Aktionen durchzuführen:

- Eine SSH-Verbindung zum Remote-Host wird hergestellt.
- Ein "conn"-Objekt wird erstellt.

  Hinweis: Wenn Sie ein Skript angeben, entweder BeanShell oder JavaScript, wird das "conn"-Objekt im Bereich dieses Skripts verfügbar gemacht.

  Sie können die öffentlichen Methoden des "conn"-Objekts im Skript optimal nutzen. Sie können diese öffentlichen Methoden verwenden, um ausgeführte Betriebsabläufe auf einem SSH-Pseudoterminal zu automatisieren. Beispiele sind das Senden von Befehlen zum Remote-SSH-Host, Warten auf die Aufforderung, nachdem alle Befehle gesendet wurden und das Abrufen der Ausgabe von jedem Befehl vom Server über SSH.

Im Gegensatz zum Operator "Telnet-Skript ausführen" stellt der Operator "SSH-Skript ausführen" nach der Anmeldung beim SSH-Host das "conn"-Objekt bereit.

Der Unterschied zwischen dem Operator "Telnet-Skript ausführen" und dem Operator "SSH-Skript ausführen" ist, dass für den Operator "SSH-Befehl ausführen" Folgendes zu beachten ist:

- Die Ausgabe von **allen** Befehlen kann automatisch im Datensatz des Operators gespeichert werden.
- Der reguläre Ausdruck, der Feld der Eingabeaufforderung für Anwender angegeben ist, wird verwendet, um nach Ausführung aller Befehle in der Liste mit der Aufforderung übereinzustimmen.
- Die "Zeit für Warten auf Aufforderungen" gilt für **alle** Befehle in der Liste. Daher sollte der Wert größer sein als die Ausführungszeit des längsten Befehls in der Liste.

Für den Operator "SSH-Skript ausführen" ist Folgendes zu beachten:

- Sie können angeben, welche Befehlsausgabe angezeigt oder im Datensatz des Operators gespeichert werden soll, indem folgende Befehle in dieser Sequenz abgerufen werden:
  1. ‘conn.sendLine()’
  2. ‘conn.waitFor()’
  3. ‘conn.getLastOutput()’
- Sie können einen anderen regulären Ausdruck angeben, um nach Ausführung aller Befehle mit der Aufforderung übereinzustimmen.
- Sie können eine andere Zeit festlegen, die nach Ausführung aller Befehle auf die Eingabeaufforderung gewartet werden soll.

Eingabeparameter

Eingabeparameter für den Operator "SSH-Skript ausführen" schließen Folgendes ein.

SSH-Skriptattribute

Name des Remote-Host

Der Hostname oder die IP des Computers, zu der eine Verbindung hergestellt werden soll.

Skripttyp:

Der Typ des Skripts, das im Feld "Inline-Skript" angegeben ist. Wählen Sie zwischen BeanShell-Skript (.bsh) und JavaScript (.js) aus. Wenn dieses Feld leer gelassen wird, verwendet der Operator standardmäßig ".bsh".

Inline-Skript?

Dieser Operator stellt zwei Methoden bereit, um das Skript anzugeben: Inline oder als Ausdruck. Aktivieren Sie dieses Kontrollkästchen, um das Skript als Inline-Skript anzugeben.

Inline-Skript

Das Skript, das in BeanShell oder JavaScript geschrieben ist, verwendet das conn-Objekt und seine API folgendermaßen:

- Ein Befehl wird an den Remote-Host gesendet
- Es wird gewartet, bis der Befehl beendet wird
- Die Ausgabe des letzten Befehls wird abgerufen

Die APIs, die das conn-Objekt anzeigt, werden in Operator "SSH-Skript ausführen" - Inline-Skript APIs (siehe Seite 112) ausführlich behandelt.

Skript als Ausdruck

Stellt das Skript als Ausdruck bereit. Informationen über das Skript selbst finden Sie im Feld "Inline-Skript".

Parameter

Die CA Process Automation-Parameter, die an das Skript übergeben werden sollen. Nur einfache CA Process Automation-Parametertypen können folgendermaßen an das Skript übergeben werden:

- PAM-Boolean wird als Boolean-Objekt übergeben.
- PAM-Date wird als Date-Objekt übergeben.
- PAM-Double wird als Double-Objekt übergeben.
- PAM-Integer wird als Integer-Objekt übergeben.
- PAM-Long wird als Long-Objekt übergeben.
- PAM-String wird als String-Objekt übergeben.
- PAM-Object Reference wird als Object Reference-Objekt übergeben.

Komplexe CA Process Automation-Parametertypen (indizierte Typen, ValueMaps usw.) können nicht an das Skript übergeben werden.

Das Skript kann auf diese Objekte über das args-Array der Objekte zugreifen, wobei args[0] dem ersten Parameter und args[1] dem zweiten Parametern in der Liste entspricht usw.

Ausgabevariablen-Namen

Die Namen der im Skript erstellten Variablen, die im Operatordatensatz gespeichert werden sollen, nachdem die Ausführung des Skripts abgeschlossen wurde.

Die Variablen müssen im Bereich des Skripts definiert sein, damit sie nach der Ausführung sichtbar sind im Datensatz des Operators gespeichert werden können.

Die Ausgabevariablen werden folgendermaßen gespeichert:

- Boolean-Objekt wird als "PAM-Boolean" gespeichert.
- Date-Objekt wird als "PAM-Date" gespeichert.
- Integer-Objekt wird als "PAM-Integer" gespeichert.
- Number-Objekt wird als "PAM-Long"- oder "PAM Double"-Objekt gespeichert.
- String-Objekt wird als "PAM-String" gespeichert.
- Character-Objekt wird als "PAM-String" gespeichert.
- Ein Array der Objekte wird als indizierter PAM-Typ gespeichert, wobei der PAM-Typ durch den Typ des ersten Objekts im Array der Objekte definiert wird.
- "Nicht definiert" wird als eine PAM-Zeichenfolge gespeichert, und zwar mit "Nicht definiert" als Wert (die Variable wurden keinem Wert zugewiesen).

Operator "SSH-Skript ausführen" - Inline-Skript APIs

Das Produkt verwendet für das Feld "Inline-Skript" des Operators "SSH-Skript ausführen" ein Skript, das in BeanShell oder JavaScript geschrieben ist. Das Skript verwendet das Objekt "conn", das die folgenden APIs darstellt:

```
void send (String str, boolean log) throws Exception
```

Diese Methode sendet Daten an den Remote-Host.

String str

Definiert die Daten, die an den Remote-Host gesendet werden sollen.

boolean log

Stellt die Daten, die an den Remote-Host gesendet werden, in den CA Process Automation-Protokollen dar bzw. blendet sie aus.

Folgen Sie für Debugging-Zwecke der Interaktion zwischen dem Operator und dem Remote-Host. Legen Sie in der Datei <*Installationsverzeichnis*>/server/conf/log4j.xml den folgenden Code fest:

```
<category name="com.company.c2o.servicegroup.netutils">


   <priority value="DEBUG" />
</category>
```

Legen Sie außerdem fest, dass die CA Process Automation-Protokolldatei (c2o.log) DEBUG-Anweisungen in "log4j.xml" akzeptiert.

Wenn Sie die Debug-Ebene festgelegt haben, beginnt der Operator "Befehlsausführung", auf der DEBUG-Ebene in die CA Process Automation-Protokolldatei (c2o.log) zu schreiben. Die CA Process Automation-Protokolle enthalten sämtliche Daten, die über "send" oder "sendLine" an den Remote-Host gesendet wurden.

Werte:

Wahr: Protokolliert den Wert "String str", wenn auf Debug-Ebene protokolliert wird.

Falsch: Protokolliert den Wert "String str" nicht.

Hinweis: "c2o.log" ist die CA Process Automation-Protokolldatei, *nicht* das Prozessprotokoll. Die Operatoren schreiben keine Meldungen in das Prozessprotokoll.

Diese Methode hat keine Rückgabewerte. Das Produkt generiert eine Ausnahme, wenn die API keine Daten im Remote-Host schreiben kann.

`public void sendLine (String str, boolean log) throws Exception`

Diese Methode hängt ein Zeichen für neue Zeile an die Daten an und sendet sie an den Remote-Host. Um zu erzwingen, dass der Remote-Host mit der Ausführung des Befehls beginnt, der im Parameter gesendet wird, verwenden Sie diese Methode.

Diese Methode hat die gleichen Parameter und Werte wie "void send (String str, boolean log) throws Exception".

`void send (String str) throws Exception`

Diese Methode entspricht "Send" (String str, "Wahr").

`void sendLine (String str) throws Exception`

Diese Methode entspricht "SendLine" (String str, "Wahr").

`public boolean waitFor(String pattern, int timeout) throws Exception`

Diese Methode liest die Ausgabe vom Remote-Host und hält an, wenn eines der folgenden Ergebnisse auftritt:

- Die Ausgabe stimmt mit dem Muster überein, das in den Parametern angegeben ist.
- Das Zeitlimit-Intervall läuft ab.

Das Produkt speichert die Ausgabe aus dem Remote-Host der einzelnen Aufrufe der Methode "waitFor" in einem Puffer, auf den über die Methode "getLastOutput()" zugegriffen werden kann. Jeder Aufruf von "waitFor" überschreibt den Pufferinhalt des vorherigen Aufrufs.

Der nächste Aufruf von "waitFor" liest die Ausgabe ab dem Punkt, an dem der vorhergehende waitFor-Aufruf zu lesen aufgehört hat. Dies ist relevant, wenn Sie die Methode mit "getLastOutput.()" verwenden. Wenn ein waitFor-Aufruf beispielsweise nicht mit der ganzen Befehlsausgabe übereinstimmt, kann der nächste Aufruf die verbleibende Ausgabe des früheren Befehls enthalten.

Wichtig! Um zu vermeiden, dass die Ausgabe des früheren Befehls mit der aktuellen Befehlsausgabe vermischt wird, rufen Sie nach jedem sendLine-Aufruf "waitFor" auf.

Die Operatoren "Telnet-Skript ausführen" und "SSH-Skript ausführen" verwenden verschiedene Mechanismen, um Daten von einem Remote-Host zu lesen:

- Bei Telnet liest "waitFor" die Daten direkt vom Host und gleicht sie während des Lesevorgangs ab.
- Bei SSH ruft "waitFor" die bisher gelesenen Daten aus einem Puffer ab und gleicht sie mit dem Muster ab.
- Wenn Sie "waitFor" nach jedem sendLine-Aufruf aufrufen, ist das Verhalten von SSH- und Telnet-Operatoren identisch.

Zu den Parametern gehören:

String pattern

Definiert einen regulären Ausdruck, mit dem die Daten, die das Produkt vom Remote-Host liest, abgeglichen werden sollen. Normalerweise steht dieses Muster für alle Daten bis zur nächsten Aufforderung (zum Beispiel ".*[$]"). Um die Befehlsausgabedaten einschließlich neuer Zeilen bis zur nächsten Aufforderung abgleichen (und abrufen) zu können, beginnen Sie das Muster mit ".*.".

Hinweis: Die Methode gleicht alle gelesenen Daten (während dieses Aufrufs von "waitFor") mit dem Muster ab. Die Methode entspricht nicht dem Muster als untergeordnete Zeichenfolge der gelesenen Daten. Ein Punkt (.) kann außerdem mit einer Zeilenendmarke übereinstimmen (er kann verwendet werden, um mit Antwortdaten, die sich über mehrere Zeilen erstrecken, übereinzustimmen).

Int timeout

Definiert das Intervall (in Sekunden), während dessen Daten vom Remote-Host gelesen und mit dem Muster abgeglichen werden sollen.

Rückgaben sind Boolesch:

Wahr

Die Daten, die bis zur Zeitüberschreitung gelesen wurden, stimmen mit dem Muster überein.

False

Die Daten, die bis zur Zeitüberschreitung gelesen wurden, stimmen nicht mit dem Muster überein.

Hinweis: Die Methode gibt einen Wert zurück, sobald eine Übereinstimmung mit dem Muster gefunden wird oder die Zeitüberschreitung erreicht ist.

In den folgenden Fällen generiert das Produkt eine Ausnahme:

- Der Parameter "Int timeout" ist kleiner oder gleich Null.
- Der Parameter "String pattern" ist ungültig oder leer.
- Das Produkt kann keine Daten aus der SSH-Sitzung lesen.
- Das Muster enthält ungültige Syntax.
- Beim Abgleichen des Musters mit den empfangenen Daten tritt ein Fehler auf.

`public String getLastOutput()`

Diese Methode gibt den Inhalt des Puffers zurück, an dem die Daten beim letzten Aufruf der Methode "waitFor" gespeichert und vom Remote-Host gelesen wurden. Diese Daten stimmen nicht immer mit dem waitFor-Muster überein. Der Puffer speichert die Werte, die durch den letzten waitFor-Aufruf gelesen wurden, unabhängig davon, ob "waitFor" "true" oder "false" zurückgegeben hat.

Diese Methode hat keine Parameter.

Diese Methode generiert keine Ausnahmen.

Informationen für Remote-Anmeldung

Pseudoterminaltyp

Der Pseudoterminaltyp, der bei SSH-Verbindungen angefordert wird. Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist. Wenn das Feld leer gelassen wird, verwendet der Operator den Standardwert, der auf der Operatorkategorieebene festgelegt ist. Wenn dieser Wert leer ist, wird der Operator standardmäßig auf VT100 festgelegt.

- VT100 funktioniert normalerweise mit den meisten Computern (besonders Linux-basiert).
- VT400 funktioniert normalerweise mit Windows-basierten Computern. VT400 ist für Windows-Plattformen erforderlich, besonders wenn die Ausgabe, die aus dem SSH-Server (Befehlsausgabe) abgerufen wird, Steuerzeichen anstatt Leerzeichen enthält. Zum Beispiel wenn in der Ausgabe "[19;1H" statt Leerzeichen enthalten ist. VT400 interpretiert die Leerzeichen für Windows richtig.

Andere Terminaltypen können verwendet werden. Stellen Sie sicher, dass Sie sie testen, bevor Sie den Operator in die Produktion verschieben. Einige Pseudoterminaltypen sind:

- dumb
- xterm
- vt220
- vt320
- gogrid

Überprüfen Sie die Installation und Konfiguration Ihres SSH-Servers für den unterstützten Pseudoterminal. Einige SSH-Server listen die unterstützten Pseudoterminal im Ordner "TermInfo" auf.

Der Pseudoterminaltyp steuert, wie Leerzeichen in der Befehlsausgabe angezeigt werden. Sie sollten diesen Operator gegen die Pseudoterminals testen, die vom SSH-Server unterstützt werden, um einen geeigneten Pseudoterminal zu finden, der die Leerzeichen entsprechend zurückgibt. Wenn die Leerzeichen nicht entsprechend zurückgegeben werden und kein Pseudoterminal, der diesen Issue beheben könnte, verfügbar ist, gehen Sie folgendermaßen vor:

- Ändern Sie die Eingabe des Operators, um die Beschränkung dieses SSH-Servers unterzubringen.
- Verwenden Sie JavaScript, um die Ausgabe der Befehle zu verfeinern bzw. zu extrahieren.

Wenn Sie einen Pseudoterminal anfordern, der nicht unterstützt wird, geben einige SSH-Server einen Fehler zurück, während andere Server den angeforderten Pseudoterminaltyp ignorieren und andere Pseudoterminaltypen verwenden. Überprüfen Sie die Protokolle des SSH-Servers für den Pseudoterminal, die verwendet werden, wenn der Operator ausgeführt wird.

Port

Der Port zum Anmelden am Remote-Host. Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist. Wenn dieses Feld leer gelassen wird, verwendet der Operator den auf Operatorkategorienebene festgelegten Standardwert. Wenn dieser Standardoperatorkategorienwert leer ist, verwendet der Operator Wert den 22.

Anwendername:

Der Anwendername, der für die Anmeldung beim Remote-Host verwendet wird. Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist. Wenn dieses Feld leer gelassen wird, verwendet der Operator den Standardwert, der auf der Operatorkategorieebene festgelegt ist.

Privaten Schlüssel für die Anmeldung verwenden?

Gibt an, ob ein privater Schlüssel für die Anmeldung am Remote-Host verwendet werden soll (statt Kennwortinformationen). Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist. Wenn dieses Feld leer gelassen wird, verwendet der Operator den Standardwert, der auf der Operatorkategorieebene festgelegt ist. Wählen Sie eine der folgenden Optionen aus:

Wahr

Fordert den Operator auf, einen privaten Schlüssel zu verwenden.

Wenn "Wahr" ausgewählt ist, werden folgende Felder aktiviert: "Eingabequelle des privaten Schlüssels", "Inline-Inhalt des privaten Schlüssels", "Privater Schlüssel als Ausdruck", "Pfad des privaten Schlüssels", "Passphrase für Schlüssel". Das Feld "Kennwort" wird deaktiviert.

False

Fordert den Operator auf, Kennwortinformationen zu verwenden.

Wenn "Falsch" ausgewählt ist, werden folgende Felder deaktiviert: "Eingabequelle des privaten Schlüssels", "Inline-Inhalt des privaten Schlüssels", "Privater Schlüssel als Ausdruck", "Pfad des privaten Schlüssels", "Passphrase für Schlüssel". Das Feld "Kennwort" wird aktiviert.

Wenn andere Werte eingegeben werden, wird der Operator aufgefordert, den Wert "Falsch" zu verwenden und aktiviert alle Felder (um den Anwender unterzubringen, der einen Ausdruck eingibt).

Kennwort

Das Kennwort, das für die Anmeldung beim Remote-Host verwendet wird. Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist. Wenn dieses Feld leer gelassen wird, verwendet der Operator den Standardwert, der auf der Operatorkategorieebene festgelegt ist.

Eingabequelle des privaten Schlüssels

Gibt an, wie der private Schlüssel angegeben werden soll. Wählen Sie eine der folgenden Optionen aus:

Inline-Inhalt

Aktiviert "Inline-Inhalt des privaten Schlüssels" und deaktiviert "Privater Schlüssel als Ausdruck" und "Pfad des privaten Schlüssels"

Ausdruck

Aktiviert "Privater Schlüssel als Ausdruck" und deaktiviert "Inline-Inhalt des privaten Schlüssels" und "Pfad des privaten Schlüssels"

Dateipfad

Aktiviert "Pfad des privaten Schlüssels" und deaktiviert "Inline-Inhalt des privaten Schlüssels" und "Privater Schlüssel als Ausdruck"

Inline-Inhalt des privaten Schlüssels

Der Inhalt des privaten Schlüssels für die Anmeldung beim Remote-Host. Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist. Wenn dieses Feld leer gelassen wird und "Eingabequelle des privaten Schlüssels" auf "Inline-Inhalt" festgelegt wird, verwendet der Operator den Standardwert, der auf der Operatorkategorieebene festgelegt ist.

Privater Schlüssel als Ausdruck

Die Datensatzvariable, die den Inhalt des privaten Schlüssels für die Protokollierung im Remote-Host enthält.

Hinweis: Stellen Sie sicher, dass die Datensatzvariable eine mehrzeilige Zeichenfolge ist.

Pfad des privaten Schlüssels

Der Pfad zum privaten Schlüssel für die Anmeldung beim Remote-Host. Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist. Wenn dieses Feld leer gelassen wird und "Eingabequelle des privaten Schlüssels" auf "Dateipfad" festgelegt wird, verwendet der Operator den Standardwert, der auf der Operatorkategorieebene festgelegt ist.

Passphrase für Schlüssel

Optionale Passphrase, um den Inhalt des privaten Schlüssels zu entsperren. Dieses Feld ist erforderlich, wenn der private Schlüssel mit einer Passphrase erstellt wurde. Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist.

Wenn "Passphrase für Schlüssel" leer ist, wird der Operator nicht automatisch aufgefordert, den Wert "Passphrase für Schlüssel" aus der Operatorkategorie zu erben. Tatsächlich wird das Feld "Passphrase für Schlüssel" folgendermaßen mit dem Feld "Inline-Inhalt des privaten Schlüssels", "Pfad des privaten Schlüssels" oder "Privater Schlüssel als Ausdruck" verbunden:

- Wenn die "Passphrase für Schlüssel" des Operators angegeben ist, wird sie durch den Operator verwendet.
- Der Operator verwendet eine leere "Passphrase für Schlüssel" (Passphrase nicht festgelegt), wenn die folgenden Bedingungen vorliegen:
  - Die "Passphrase für Schlüssel" des Operators ist leer
  - "Inline-Inhalt des privaten Schlüssels" wird angegeben (nicht leer)
  - "Eingabequelle des privaten Schlüssels" wird auf "Inline-Inhalt" festgelegt

- Der Operator verwendet eine leere "Passphrase für Schlüssel" (Passphrase nicht festgelegt), wenn die folgenden Bedingungen vorliegen:
  - Die "Passphrase für Schlüssel" des Operators ist leer
  - "Pfad des privaten Schlüssels" wird angegeben (nicht leer)
  - "Eingabequelle des privaten Schlüssels" wird auf "Dateipfad" festgelegt
- Der Operator verwendet eine leere "Passphrase für Schlüssel" (Passphrase nicht festgelegt), wenn die folgenden Bedingungen vorliegen:
  - Die "Passphrase für Schlüssel" des Operators ist leer
  - "Privater Schlüssel als Ausdruck" wird angegeben (nicht leer)
  - Die "Eingabequelle des privaten Schlüssels" wird auf "Ausdruck" festgelegt

Für alle anderen Fälle verwendet der Operator "SSH-Skript ausführen" die Option "Standardmäßige Passphrase für Schlüssel".

Hinweis: Die Erstellung von privaten bzw. öffentlichen Schlüsseln für SSH wird im CA Process Automation-*Handbuch für Inhaltsadministratoren* beschrieben.

Ausgabeparameter

Alle Variablen in der Liste "Ausgabevariablen-Namen" werden mit dem entsprechenden CA Process Automation-Typ erstellt.

Wenn ein Variablenname in "Ausgabevariablen-Namen" nicht im Skript vorhanden ist, erstellt der Operator die entsprechende Variable als leere Zeichenfolge.

Der BeanShell-Interpreter stellt eine robuste Umgebung bereit. Wenn das Skript in dieser Umgebung eine Ausnahme ausgibt oder einen Fehler enthält, können die Variablen, die vor dem Auftreten des Fehlers definiert und im Skript initialisiert wurden, mit ihren Werten abgerufen werden. Andererseits können mit dem JavaScript-Interpreter keine Variablen mit ihrem eigenen Wert abgerufen werden, wenn das Skript eine Ausnahme ausgibt oder einen Fehler enthält.

Zu den Ausgabeparametern gehören:

SSHHost

inLineScriptLanguage

inlineScriptType

inLineScript

scriptExpression

parameters

outputVariables

SSHPseudoTerminalType

SSHPort

SSHUsername

SSHUsePrivateKey

SSHPassword

SSHPrivateKeyOption

SSHPrivateKey

SSHPrivateKeyVar

SSHPrivateKeyPath

SSHPrivateKeyPassphrase

Operator-Ports

Erfolg

Der Operator wurde erfolgreich abgeschlossen.

Fehlgeschlagen

Der Operator schlägt aus folgenden Gründen fehl:

- Der Anwender gibt ungültige Eingabe an. Zum Beispiel: Leeres Inline-Skript, leerer Remote-Host, negativer Port, leerer Anwendername usw.
- Der Anwender gibt einen anderen Inline-Skript-Typ als ".bsh" und ".js" an.
- Es kann keine Verbindung zum Remote-Host hergestellt werden. Überprüfen Sie den im angegebenen Operator Remote-Host und Port.
- Der Anwender gibt einen unbekannten Host im Remote-Host an.
- Der Anwender kann auf dem Remote-Host nicht authentifiziert werden:
  - Anwender-/Kennwortauthentifizierung: Anwendername oder Kennwort ist ungültig.

- Anwender-/private Schlüsselauthentifizierung:

  - Eine ungültige Passphrase wird für einen passphrasengeschützten privaten Schlüssel angegeben: Der Operator zeigt an, dass der Schlüssel aufgrund eines internen E-/A-Problems nicht gelesen werden kann

  - Es fehlt die Passphrase für einen passphrasengeschützten privaten Schlüssel: Der Operator zeigt an, dass die erforderliche Passphrase für Schlüsselauthentifizierung nicht angegeben ist.

  - Ein ungültiger privater Schlüssel wird angegeben: Der Operator zeigt an, dass der private Schlüssel nicht analysiert werden kann, das Schlüsselformat nicht unterstützt wird oder der Schlüssel aufgrund eines internen E-/A-Problems nicht gelesen werden kann.

  - Der Pfad zur privaten Schlüsseldatei ist ungültig, existiert nicht oder kann nicht gelesen werden.

- Der angegebene Anwendername bzw. das angegebene Kennwort sind nicht berechtigt, sich über SSH anzumelden.
- Der Anwender gibt einen komplexen Datentyp in der Liste von Parametern an. Komplexe Werttypen (Arrays, ValueMaps usw.) können nicht an das Skript übergeben werden. Verwenden Sie einfache Werttypen wie "Doppelt", "Ganzzahl", "Lang", "Zeichenfolge", "Datum" und "Boolesch".
- Beim Ausführen eines ".bsh"- oder ".js"-Skripts:
  - Beim Auswerten des Skripts tritt ein Analyse- oder Syntaxfehler auf.
  - Das Skript gibt eine Ausnahme aus.
  - Beim Ausführen des Skripts tritt ein Fehler auf.
  - Beim Abrufen einer Variablen aus dem Skriptbereich tritt ein Fehler auf (am Ende der Ausführung).

Anwenderspezifische Ports

Wenn während des Prozessdesigns durch den Anwender festgelegt.

Beispiel

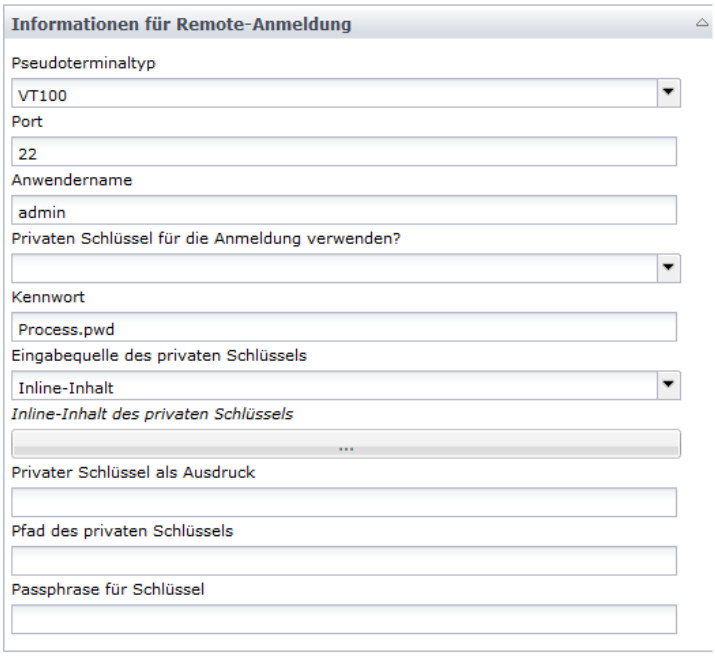
Die folgende Prozedur ist ein Beispiel, wie der Operator "SSH-Skript ausführen" verwendet wird.

Gehen Sie folgendermaßen vor::

1. Der Operator "SSH-Skript ausführen" liest die von Ihnen angegebenen Anmeldeinformationen und verwendet sie für folgende Aktionen:

   - Um eine Verbindung zum Remote-SSH-Host herzustellen und um sich beim Remote-SSH-Host zu authentifizieren.
   - Um das "conn"-Objekt zu erstellen.

   Es folgt ein Beispiel für den ausgefüllten Bereich "Informationen für Remote-Anmeldung" für diesen Operator:



2. Führen Sie die Skriptparameter folgendermaßen durch:

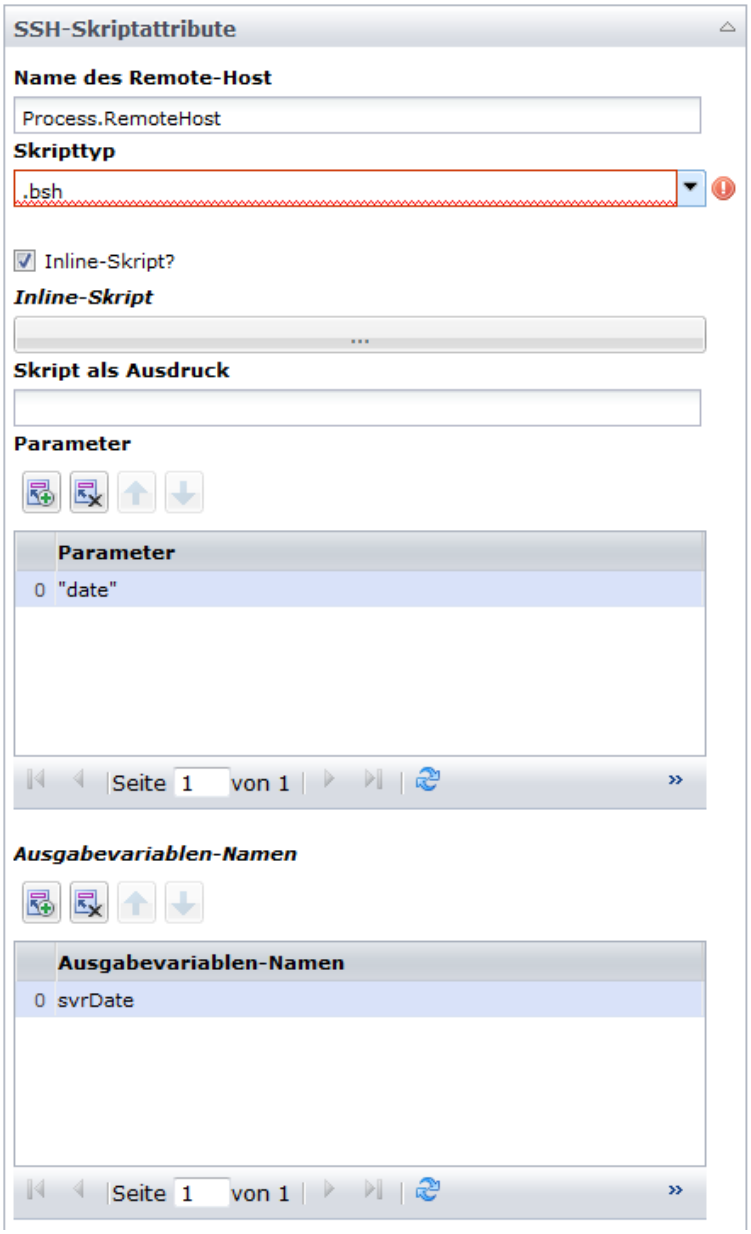
   a. Geben Sie den Namen des Remote-Host an.

   b. Geben Sie die Parameter an, die an das Skript übergeben werden sollen. Es folgen die Werte für dieses Beispiel:

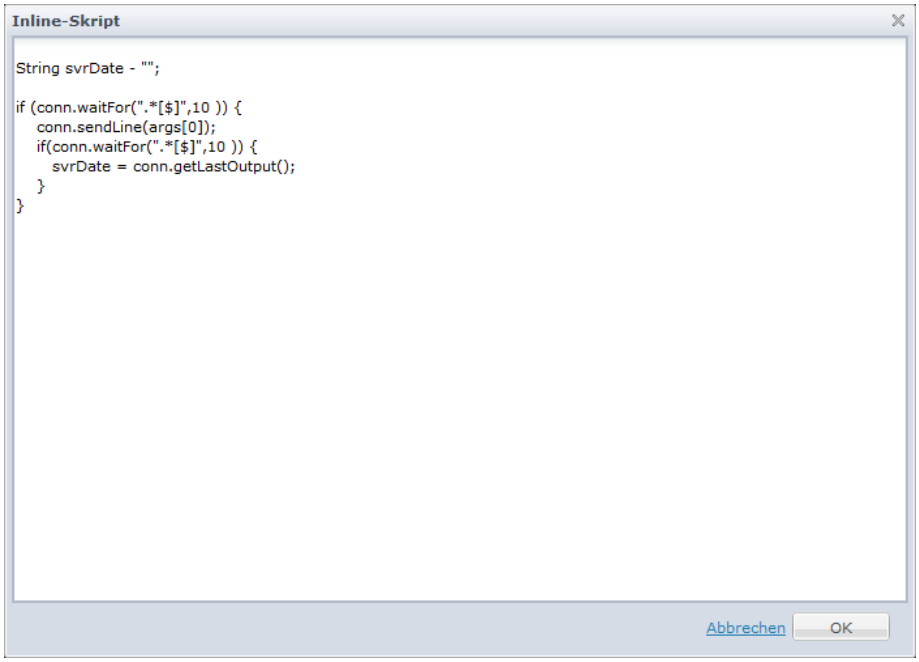
      Eine Zeichenfolge, deren Wert das Wort "Datum" ist.

   c. Geben Sie den Namen der Ausgabevariablen an, die Sie im Skript erstellt haben (BeanShell oder JavaScript) und die im Datensatz des Operators gespeichert werden sollen, nachdem die Ausführung abgeschlossen wurde (hier, "svrDate").

      Hinweis: Die Erstellung von Datensatzvariablen direkt aus dem Skript wird derzeit nicht unterstützt.


SSH-Skriptattribute
Name des Remote-Host
Process.RemoteHost
Skripttyp
.bsh
Inline-Skript?
Inline-Skript
...
Skript als Ausdruck
Parameter
Parameter
0 "date"
Seite 1 von 1
Ausgabevariablen-Namen
Ausgabevariablen-Namen
0 svrDate
Seite 1 von 1

3. Im Inline-Skript können Sie das "conn"-Objekt optimal nutzen, wie im folgenden BeanShell-Skriptbeispiel dargestellt:



   a. Erstellen Sie die Variable "svrDate", die im Skriptbereich sichtbar sein soll, sodass sie nach der Ausführung im Datensatz des Operators gespeichert werden kann.

   b. Verwenden Sie "conn.waitFor()", um bis zu 10 Sekunden auf die erste Aufforderung ".*[$]" (RegEx) zu warten.

   c. Wenn die Aufforderung innerhalb von 10 Sekunden gefunden wird, verwenden Sie "conn.sendLine()", um den Wert des ersten Parameters, das an das Skript übergeben wurde, zu senden, gefolgt von einem neuen Zeilenendzeichen. In diesem Beispiel ist der Wert: args[0] = das Wort "date".

   d. Verwenden Sie "conn.waitFor()", um bis zu 10 Sekunden auf die nächste Aufforderung ".*[$]" (RegEx) zu warten.

   e. Wenn die Aufforderung innerhalb 10 Sekunden gefunden wird, verwenden Sie "conn.getLastOutput()", um die Ausgabe abzurufen, die während des letzten Aufrufs der Methode "waitFor" gelesen und in "svrDate" gespeichert wurde.

Nach der Ausführung speichert der Operator das Zeichenfolgenobjekt "svrDate" als CA Process Automation-Zeichenfolge im Datensatz des Operators.

Hinweis: Die übereinstimmende Aufforderung wird in der Ausgabe angezeigt, die von "conn.getLastOutput()" zurückgegeben wird. Einige SSH-Server geben diese Aufforderung in der Ausgabe zweimal zurück, während andere sie nur einmal zurückgeben.

Operator "Telnet-Befehl ausführen"

Der Operator "Telnet-Befehl ausführen" führt folgende Aktionen aus:

- Öffnet eine Telnet-Verbindung zum Remote-Host.
- Sendet gleichzeitig einen Befehl.
- Liest die Ausgabe des Befehls, bis die Eingabeaufforderung angezeigt wird, um anzuzeigen, dass der Befehl abgeschlossen wurde.
- Sendet den nächsten Befehl.

Hinweis: Der Operator "Telnet-Befehl ausführen" und der Operator "SSH-Befehl ausführen" melden sich anders am Remote-Host an. Der Operator "Telnet-Befehl ausführen" führt eine interaktive Anmeldung aus.

Sie können die Zeit festlegen, die maximal auf die Eingabeaufforderung gewartet werden soll, bevor der Operator fehlschlägt. Stellen Sie sicher, dass diese Einstellung größer ist als die Ausführungszeit des längsten Befehls, den dieser Operator ausführen kann.

Sie können festlegen, dass der Operator "Telnet-Befehl ausführen" nach Anmeldung und bevor der Befehl ausgeführt wird, zu einem anderen Anwender (einschließlich "Root") wechselt. Das Wechseln von Anwendern ermöglicht es, dass die nachfolgenden Befehle unter einem anderen Anwender ausgeführt werden. Das Wechseln zu anderen Anwendern wird interaktiv durchgeführt.

Eingabeparameter

Hinweise:

- Für alle Eingaben, die im Operator "Telnet-Befehl ausführen" als regulärer Ausdruck angegeben werden können, gilt Folgendes:
  - Der Operator stimmt die gesamten Antwortdaten mit dem Muster ab.
  - Der Operator entspricht nicht dem Muster als untergeordnete Zeichenfolge der Antwortdaten.
- Ein Punkt '.' entspricht einem neuen Zeilenabschluss (kann verwendet werden, um mehrzeilige Antwortdaten abzustimmen).

Befehle

Name des Remote-Host

Der Hostname oder die IP des Computers, zu der eine Verbindung hergestellt werden soll.

Indizierte Zeichenfolgenvariable für Befehle verwenden?

Wenn dieses Kontrollkästchen nicht aktiviert ist, können Sie Befehle in das Feld "Befehle" eingeben. Wählen Sie diese Option aus, um die Befehle als indizierte Zeichenfolgenvariablen im Feld "Befehle - Indizierte Zeichenfolgenvariable" anzugeben.

Befehle

Liste der Befehle, die auf dem Remote-Host ausgeführt werden sollen. Beenden Sie die Liste nicht mit dem Befehl "exit", da der Operator die Telnet-Sitzung automatisch verlässt, nachdem der letzte Befehl ausgeführt wurde.

Befehle - Indizierte Zeichenfolgenvariable

Der Name der Datensatzvariable, die eine Liste von Befehlen enthält, die auf dem Remote-Host ausgeführt werden sollen. Beenden Sie die Liste nicht mit dem Befehl "exit". Der Operator verlässt automatisch die Telnet-Sitzung, nachdem der letzte Befehl ausgeführt wurde.

Ausgabe in Datensatzvariable speichern?

Aktivieren Sie dieses Kontrollkästchen, um die Ausgabe aller Befehle in den Datensatz des Operators zu kopieren. Die Ausgabe von jedem Befehl ist in der Variable "TelnetCommandsOutput" gespeichert.

Größenbeschränkung für die Datensatzvariable der Befehlsausgabe (Byte)

Gibt die Höchstanzahl von Byte der Befehlsausgabe an, die jeweils in der Datensatzvariable des Operators gespeichert werden soll. Wenn diese Zahl nicht angegeben ist, verwendet der Operator den Wert 4096.

Informationen für Remote-Anmeldung

Pseudoterminaltyp

Der Pseudoterminaltyp, der bei Telnet-Verbindungen angefordert wird. Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist. Wenn das Feld leer gelassen wird, verwendet der Operator den Standardwert, der auf der Operatorkategorieebene festgelegt ist. Wenn dieser Wert leer ist, wird der Operator standardmäßig auf VT100 festgelegt.

- VT100 funktioniert normalerweise mit den meisten Computern (besonders Linux-basiert).
- VT400 funktioniert normalerweise mit Windows-basierten Computern. VT400 ist für Windows-Plattformen erforderlich, besonders wenn die Ausgabe, die aus dem Telnet-Server (Befehlsausgabe) abgerufen wird, Steuerzeichen anstatt Leerzeichen enthält. Zum Beispiel wenn in der Ausgabe "[19;1H" statt Leerzeichen enthalten ist. VT400 interpretiert die Leerzeichen für Windows richtig.

Andere Terminaltypen können verwendet werden. Stellen Sie sicher, dass Sie sie testen, bevor Sie den Operator in die Produktion verschieben. Einige Pseudoterminaltypen sind:

- dumb
- xterm
- vt220
- vt320
- gogrid

Überprüfen Sie die Installation und Konfiguration Ihres Telnet-Servers für den unterstützten Pseudoterminal.

Der Pseudoterminaltyp steuert, wie Leerzeichen in der Befehlsausgabe angezeigt werden. Sie sollten diesen Operator gegen die Pseudoterminals testen, die vom Telnet-Server unterstützt werden, um einen geeigneten Pseudoterminal zu finden, der die Leerzeichen entsprechend zurückgibt. Wenn die Leerzeichen nicht entsprechend zurückgegeben werden und kein Pseudoterminal, der diesen Issue beheben könnte, verfügbar ist, gehen Sie folgendermaßen vor:

- Ändern Sie die Eingabe des Operators, um die Beschränkung dieses Telnet-Servers unterzubringen.
- Verwenden Sie JavaScript, um die Ausgabe der Befehle zu verfeinern bzw. zu extrahieren.

Wenn Sie einen Pseudoterminal anfordern, der nicht unterstützt wird, geben einige Telnet-Server einen Fehler zurück, während andere Server den angeforderten Pseudoterminaltyp ignorieren und andere Pseudoterminaltypen verwenden. Überprüfen Sie die Protokolle des Telnet-Servers für den Pseudoterminal, die verwendet werden, wenn der Operator ausgeführt wird.

Remote-Port

Der Port zum Verbinden mit dem Remote-Host. Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist. Wenn dieses Feld leer gelassen wird, verwendet der Operator den Standardwert, der auf der Operatorkategorieebene festgelegt ist. Wenn dieser standardmäßige Kategoriewert leer ist, verwendet der Operator den Wert: 23.

Zeitlimit für Verbindung (Sek.)

Das Verbindungszeitlimit, nach dessen Ablauf der Verbindungsversuch abgebrochen wird. Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist. Wenn dieses Feld leer gelassen wird, verwendet der Operator den Standardwert, der auf der Operatorkategorieebene festgelegt ist. Wenn dieser standardmäßige Kategoriewert leer ist, verwendet der Operator den Wert: 20.

Anmeldeschema

Wählen Sie für das Anmeldeschema einen der folgenden Werte aus:

0

Fordert den Operator auf, Anwendernamen und Kennwort zu verwenden

1

Fordert den Operator auf, nur das Kennwort zu verwenden, wodurch folgende Felder deaktiviert werden: "Textaufforderung für Kennwort" und "Kennwort"

2

Fordert den Operator auf, keinen Anwendernamen und kein Kennwort zu verwenden, wodurch folgende Felder deaktiviert werden: "Textaufforderung für Anwenderanmeldung", "Anwendername", "Textaufforderung für Anwenderkennwort" und "Kennwort"

Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist. Wenn Sie keinen Wert angeben und das Feld leer lassen, verwendet der Operator den Wert, der auf der Operatorkategorieebene festgelegt wurde. Jeder andere Wert fordert den Operator auf, "Anwendernamen" und "Kennwort" zu verwenden

Textaufforderung für Anwenderanmeldung

Die Textaufforderung, die anzeigt, dass der Remote-Host eine Anmelde-ID für das Anmelden benötigt. Dies ist im Allgemeinen:

- Anmeldung:

  oder

- anmeldung:

Dieser Parameter muss als regulärer Ausdruck angegeben werden. Zum Beispiel, ".*nmeldung: ", um alle Eingaben zu finden, auf die "nmeldung: " folgt (einschließlich neuer Zeilen).

Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist. Wenn dieses Feld leer gelassen wird, verwendet der Operator den Standardwert, der auf der Operatorkategorieebene festgelegt ist.

Anwendername:

Der Anwendername, der für die Anmeldung beim Remote-Host verwendet wird. Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist. Wenn dieses Feld leer gelassen wird, verwendet der Operator den Standardwert, der auf der Operatorkategorieebene festgelegt ist.

Textaufforderung für Anwenderkennwort

Die Textaufforderung, die anzeigt, dass der Remote-Host ein Kennwort für das Anmelden des Anwenders benötigt. Dies ist im Allgemeinen:

- Kennwort:

  oder

- kennwort:

Dieser Parameter muss als regulärer Ausdruck angegeben werden. Zum Beispiel, ".*ennwort: ", um alle Eingaben zu finden, auf die "ennwort: " folgt (einschließlich neuer Zeilen). Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist. Wenn dieses Feld leer gelassen wird, verwendet der Operator den Standardwert, der auf der Operatorkategorieebene festgelegt ist.

Kennwort

Das Kennwort, das für die Anmeldung beim Remote-Host verwendet wird. Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist. Wenn dieses Feld leer gelassen wird, verwendet der Operator den Standardwert, der auf der Operatorkategorieebene festgelegt ist.

Eingabeaufforderung für Anwender

Dieses Feld erfüllt zwei Zwecke:

- Um anzuzeigen, dass der Anwender angemeldet ist.
- Um anzuzeigen, dass ein Befehl in der Liste der Befehle oder Befehls-Array, der in der Telnet-Sitzung unter diesem ausgeführt wurde, beendet wurde, und dass der Operator den nächsten Befehl in der Liste senden kann.

Dieses Feld ist im Allgemeinen eine Anzeige der Eingabeaufforderung des Anwenders. Es ist im Allgemeinen "#", $, ">" usw. Dieses Feld muss als regulärer Ausdruck angegeben werden. Zum Beispiel: ".*[$>?:#]", um alle Eingaben (einschließlich neuer Zeilen) zu finden, auf die "$", ">", "?", ":" oder "#" folgt. Sie sollten alle Aufforderungen angeben, die während der Ausführung der Befehle angezeigt werden sollen. Der regulären Ausdruck sollte mit .* beginnen, um alle Daten finden zu können, die durch den Befehl zurückgegeben wurden, bis die Aufforderung angezeigt wird. Dieser reguläre Ausdruck sollte bis zur nächsten Aufforderung alle Ausgaben des Befehls finden können.

Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist. Wenn dieses Feld leer gelassen wird, verwendet der Operator den Standardwert, der auf der Operatorkategorieebene festgelegt ist.

Hinweis: Die Klammern um "$" sind erforderlich, um das $-Zeichen anzuzeigen. "$" hat eine in regulären Ausdrücken eine besondere Bedeutung, wenn es nicht in Klammern gesetzt wird.

Zeit für Warten auf Aufforderungen

Die Zeit (in Sekunden), während der auf eine Aufforderung zum Senden von Befehlen gewartet wird. Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist. Wenn dieses Feld leer gelassen wird, verwendet der Operator den Standardwert, der auf der Operatorkategorieebene festgelegt ist. Wenn dieser standardmäßige Wert leer ist, verwendet der Operator den Wert 60.

Dieses Feld gilt für Aufforderungen, die erwartet werden, nachdem alle Befehle in der Anmeldung und Befehle für Anwenderwechsel ausgeführt wurden. Es gilt außerdem für Aufforderungen, die erwartet werden, nachdem alle im Operator angegebenen Befehle ausgeführt wurden. Der Operator weiß nicht, ob ein in der Telnet-Sitzung ausgeführter Befehl alle Daten zurückgegeben hat. Daher fährt der Operator mit dem Lesen der Ausgabe des Befehls fort, bis eine Übereinstimmung mit dem angegebenen Anwender oder mit einer Eingabeaufforderung für Anwenderwechsel vorhanden ist oder bis dieses Zeitlimit abgelaufen ist (je nachdem, was zuerst zutrifft). Der Operator fährt dann mit der Verarbeitung der Ausgabe des Befehls fort, bevor zum nächsten Befehl gewechselt wird oder der Operator fehlschlägt.

Wichtig! Stellen Sie sicher, dass diese Einstellung größer ist als die Ausführungszeit des längsten Befehls, der durch diesen Operator ausgeführt werden soll.

Kennwort zu Anwenderwechsel

Befehl bzw. Skript als anderer Anwender ausführen?

Sollte das Skript oder die angegebenen Befehle als ein anderer Anwender ausgeführt werden? Wählen Sie "Wahr" oder "Falsch" aus.

- Wenn "Wahr", wechselt der derzeit angemeldete Anwender zu einem anderen Anwender, bevor er die Befehle ausführt.
- Wenn "Falsch" werden folgende Felder deaktiviert: "Befehl für Anwenderwechsel", "Textaufforderung für Kennwort bei Anwenderwechsel", "Kennwort für Anwenderwechsel" und "Eingabeaufforderung für Anwenderwechsel".

Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist. Wenn dieses Feld leer gelassen wird, verwendet der Operator den Standardwert, der auf der Operatorkategorieebene festgelegt ist.

Befehl für Anwenderwechsel

Der Befehl, um den Anwender auf dem Remote-Host zu wechseln. Dies ist im Allgemeinen:

- su - Anwendername

  oder

- sudo su - Anwendername

Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist. Wenn dieses Feld leer gelassen wird, verwendet der Operator den Standardwert, der auf der Operatorkategorieebene festgelegt ist.

Textaufforderung für Kennwort bei Anwenderwechsel

Die Textaufforderung, die anzeigt, dass der Remote-Host für das Wechseln zwischen verschiedenen Anwendern ein Kennwort benötigt. Dies ist im Allgemeinen:

- Kennwort:

  oder

- kennwort:

Dieser Parameter muss als regulärer Ausdruck angegeben werden. Zum Beispiel, ".*ennwort: ", um alle Eingaben zu finden, auf die "ennwort: " folgt (einschließlich neuer Zeilen).

Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist. Wenn dieses Feld leer gelassen wird, verwendet der Operator den Standardwert, der auf der Operatorkategorieebene festgelegt ist.

Wenn für das Wechseln zwischen verschiedenen Anwendern kein Kennwort erforderlich ist (zum Beispiel, wenn man von Root zu einem anderen Anwender gewechselt wird), können Sie einen beliebigen Wert in dieses Feld eingeben. Der Operator versucht die Daten zu finden, die aus der Telnet-Sitzung gelesen wurden, nachdem zunächst "Eingabeaufforderung für Anwenderwechsel" gegen "Textaufforderung für Kennwort bei Anwenderwechsel" gesendet wurde. Wenn keine Übereinstimmung vorhanden ist, wird versucht, die Daten mit der Eingabeaufforderung für Anwenderwechsel abzustimmen, um zu überprüfen, ob ein Kennwort erforderlich ist.

Kennwort für Anwenderwechsel

Das Kennwort zum Wechseln von einem Anwender zu einem anderen. Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist. Wenn dieses Feld leer gelassen wird, verwendet der Operator den Standardwert, der auf der Operatorkategorieebene festgelegt ist.

Dieses Feld wird nicht verwendet, wenn für das Wechseln zu einem anderen Anwender kein Kennwort erforderlich ist.

Eingabeaufforderung für Anwenderwechsel

Dieses Feld erfüllt zwei Zwecke:

- Um anzuzeigen, dass der Anwenderwechsel stattgefunden hat.
- Um anzuzeigen, dass ein Befehl in der Liste der Befehle oder Befehls-Array, der in der Telnet-Sitzung unter dem neuen Anwender (zu dem der Operator wechselt) ausgeführt wurde, beendet wurde, und dass der Operator den nächsten Befehl in der Liste senden kann.

Dieses Feld ist im Allgemeinen eine Anzeige der Eingabeaufforderung des neuen Anwenders (zu dem der Operator wechselt). Es ist im Allgemeinen "#", $, ">" usw. Dieses Feld muss als regulärer Ausdruck angegeben werden.

Zum Beispiel: ".*[$>?:#]", um alle Eingaben (einschließlich neuer Zeilen) zu finden, auf die "$", ">", "?", ":" oder "#" folgt. Geben Sie alle Aufforderungen an, die während der Ausführung der Befehle angezeigt werden sollen. Der regulären Ausdruck sollte auch mit .* beginnen, um alle Daten zu finden, die durch den Befehl zurückgegeben wurden, bis die Aufforderung angezeigt wird. Dieser reguläre Ausdruck sollte bis zur nächsten Aufforderung alle Ausgaben des Befehls finden können.

Hinweis: Die Klammern um "$" sind erforderlich, um das $-Zeichen anzuzeigen. "$" hat eine in regulären Ausdrücken eine besondere Bedeutung, wenn es nicht in Klammern gesetzt wird.

Seien Sie vorsichtig mit den RegEx, um falsche Positive zu vermeiden, wie z. B.:

Der Anwender gibt ein ungültiges Kennwort ein, wenn er zum Root wechselt:

```
# su – root
Kennwort:
```

Die Antwort für ein ungültiges Kennwort endet mit #:

```
su: Sorry
#
```

Gibt die gleiche Aufforderung, wie wenn der Anwender ein gültiges Kennwort eingibt, wo die Antwort auch mit # endet:

```
Sun Microsystems Inc.   SunOS 5.10      Generic January 2005
#
```

Dieses Feld überschreibt den Wert, der auf der Operatorkategorieebene angegeben ist. Wenn dieses Feld leer gelassen wird, verwendet der Operator den Standardwert, der auf der Operatorkategorieebene festgelegt ist.

Ausgabeparameter

TelnetUserLoginPrompt

Die tatsächliche "Textaufforderung für Anwenderanmeldung", die aus der SSH-Sitzung innerhalb der "Zeit für Warten auf Aufforderungen" gelesen wurde. Dies sind die Daten, die mit den Mustern abgeglichen werden, die im Feld "Textaufforderung für Anwenderanmeldung" angegeben sind.

TelnetUserPasswordPrompt

Die tatsächliche "Textaufforderung für Anwenderkennwort", die aus der SSH-Sitzung innerhalb der "Zeit für Warten auf Aufforderungen" gelesen wurde. Dies sind die Daten, die mit den Mustern abgeglichen werden, die im Feld "Textaufforderung für Anwenderkennwort" angegeben sind.

TelnetUserCommandPrompt

Die tatsächliche "Eingabeaufforderung für Anwender", die das erste Mal aus der Telnet-Sitzung (innerhalb der "Zeit für Warten auf Aufforderungen"), entweder nach der Anmeldung oder nach der Verbindung, gelesen wurde (wenn keine Anmeldung erforderlich ist). Dies sind die Daten, die mit den Mustern abgeglichen werden, die im Feld "Eingabeaufforderung für Anwender" angegeben sind.

TelnetSwitchUserPasswordPrompt

Die tatsächliche "Textaufforderung für Kennwort bei Anwenderwechsel", die aus der SSH-Sitzung innerhalb der "Zeit für Warten auf Aufforderungen" gelesen wurde. Dies sind die Daten, die mit den Mustern abgeglichen werden, die im Feld "Textaufforderung für Kennwort bei Anwenderwechsel" angegeben sind.

Dieses Feld ist leer, wenn zum Wechseln zwischen Anwendern kein Kennwort erforderlich ist, wie z. B. wenn von Root zu einem anderen Anwender gewechselt wird oder wenn der Operator nicht zum Wechseln zwischen Anwendern eingerichtet ist.

TelnetSwitchUserCommandPrompt

Die tatsächliche "Eingabeaufforderung für Anwenderwechsel", die das erste Mal aus der Telnet-Sitzung (innerhalb "Zeit für Warten auf Aufforderungen") gelesen wurde, nachdem zum neuen Anwender gewechselt wurde. Dies sind die Daten, die mit den Mustern abgeglichen werden, die im Feld "Eingabeaufforderung für Anwenderwechsel" angegeben sind. Dieses Feld ist leer, wenn der Operator nicht zum Wechseln zwischen Anwendern eingerichtet ist.

TelnetCommandsOutput

Eine indizierte Zeichenfolge mit der Ausgabe, die aus jedem Befehl (innerhalb der "Zeit für Warten auf Aufforderungen") gelesen wurde, der auf dem Remote-Host ausführt wurde. Die Ausgabe für jeden Befehl ist auf den Wert gekürzt, der in der Größenbeschränkung für die Befehlsausgabe angegeben ist.

Die vollständige Ausgabe von jedem Befehl enthält die nächste Aufforderung.

TelnetHost

CommandsAsAnArray

parameters

CommandsArray

isPostToOutVar

PostToOutVarSizeLimit

TelnetPseudoTerminalType

TelnetPort

TelnetHostConnectTimeout

TelnetloginScheme

TelnetHostUserLoginPrompt

TelnetUsername

TelnetHostUserPasswordPrompt

TelnetPassword

TelnetHostUserCommandPrompt

TelnetHostWaitTimeout

TelnetSwitchUser

TelnetHostSwitchUserCommand

TelnetHostSwitchUserPasswordPrompt

TelnetHostSwitchUserPassword

TelnetHostSwitchUserCommandPrompt

TelnetSwitchUserPasswordPrompt

TelnetSwitchUserCommandPrompt

TelnetUserLoginPrompt

TelnetUserPasswordPrompt

TelnetUserCommandPrompt

Operator-Ports

Erfolg

- Der Operator wurde erfolgreich abgeschlossen.
- Der Operator hängt von den Mustern ab, die in "Eingabeaufforderung für Anwender" und "Eingabeaufforderung für Anwenderwechsel" angegeben sind, um zu entscheiden, ob ein Befehl (und der Operator) erfolgreich ist oder fehlschlägt.

  Zum Beispiel führt das Muster ".*#" zu folgendem Befehl: "badCmd"

  1. Der Operator führt den Befehl aus:

     ```
     # badCmd
     ```

  2. Die Ausgabe wird gelesen (die mit der Aufforderung ".*#" übereinstimmt):

     ```
     badCmd:  not found
     #
     ```

  3. Führen Sie den nächsten Befehl aus:

Fehlgeschlagen

- Der Anwender gibt ungültige Eingaben an. Zum Beispiel negativer Remote-Port, negatives Zeitlimit für Verbindung, 0 oder negatives Aufforderungs-Zeitlimit, leere Eingabeaufforderung für Anwender usw.
- Es konnte kein Anmeldung durchgeführt werden. Die Daten, die aus der Telnet-Sitzung (innerhalb der "Zeit für Warten auf Aufforderungen") gelesen wurden, konnten nicht mit dem angegebenen Muster "Textaufforderung für Anwenderanmeldung" abgestimmt werden.
- Es konnte kein Anmeldung durchgeführt werden. Die Daten, die aus der Telnet-Sitzung (innerhalb der "Zeit für Warten auf Aufforderungen") gelesen wurden, konnten nicht mit dem angegebenen Muster "Textaufforderung für Anwenderkennwort" abgestimmt werden.
- Es konnte kein Anmeldung durchgeführt werden. Die Daten, die aus der Telnet-Sitzung (innerhalb der "Zeit für Warten auf Aufforderungen") gelesen wurden, konnten nicht mit dem angegebenen Muster ">Eingabeaufforderung für Anwender" abgestimmt werden.
- Es kann nicht zu einem anderen Anwender gewechselt werden. Die Daten, die aus der Telnet-Sitzung (innerhalb der "Zeit für Warten auf Aufforderungen") gelesen wurden, konnten nicht mit dem angegebenen Muster "Eingabeaufforderung für Anwenderwechsel" abgestimmt werden.
- Es kann nicht zu einem anderen Anwender gewechselt werden. Die Daten, die aus der Telnet-Sitzung (innerhalb der "Zeit für Warten auf Aufforderungen") gelesen wurden, konnten nicht mit dem angegebenen Muster "Textaufforderung für Kennwort bei Anwenderwechsel" abgestimmt werden.

- Der Anwender gibt ungültige Anmeldeinformationen für den Anwendernamen bzw. für das Kennwort an.
- Der Anwender gibt für das Wechseln des Anwenders ungültige Anmeldeinformationen für den Anwendernamen bzw. für das Kennwort an.
- Es kann nicht zu einem anderen Anwender gewechselt werden, sofern nicht das Schema des Anwendernamens bzw. des Kennworts für die Anmeldung in der Telnet-Sitzung verwendet wird.
- Der Anwender gibt einen unbekannten Remote-Host an.
- Die Telnet-Verbindung zum Remote-Host wird abgelehnt (Telnet ist nicht zulässig).
- Der angegebene Anwendername bzw. das angegebene Kennwort sind nicht berechtigt, sich über Telnet anzumelden.
- Der Operator konnte die Eingabeaufforderung (in "Eingabeaufforderung für Anwender" oder "Eingabeaufforderung für Anwenderwechsel" angegeben) nicht mit der Ausgabe abstimmen, die innerhalb der "Zeit für Warten auf Aufforderungen" aus einem Befehl gelesen wurde. In diesem Fall entweder:
  - Der Befehl hat die Ausführung fertiggestellt und Daten zurückgegeben, die nicht mit dem Muster in "Eingabeaufforderung für Anwender" oder "Eingabeaufforderung für Anwenderwechsel" übereinstimmen.
  - Die Ausführung des Befehls dauert länger als die "Zeit für Warten auf Aufforderungen", und der Operator hält das Lesen der Ausgabedaten an, bevor der Befehl beendet wird. Dadurch wurden die gelesenen Daten nicht mit dem Muster in "Eingabeaufforderung für Anwender" oder "Eingabeaufforderung für Anwenderwechsel" abgestimmt. In diesem Fall muss der Wert in "Zeit für Warten auf Aufforderungen" größer sein als die Ausführungszeit des längsten Befehls in der Liste.

  Hinweis: In diesem Fall werden nachfolgende Befehle in der Liste nicht ausgeführt, und "TelnetCommandsOutput" enthält nur die Ausgabe der Befehle, die vorher ausgeführt wurden (einschließlich der Ausgabe, die für den aktuellen Befehl gelesen wurde).

Anwenderspezifische Ports

Wenn während des Prozessdesigns durch den Anwender festgelegt.

Beispiel

Interaktive Verwendung des Operators "Telnet-Befehl ausführen"

Der Operator "Telnet-Befehl ausführen" funktioniert folgendermaßen auf interaktiver Weise:

1. Stellen Sie eine Telnet-Verbindung zum Remote-Host her.
2. Führen Sie einen der folgenden Schritte aus:
   - Wenn keine Anmeldung erforderlich ist, fahren Sie mit Schritt 8 fort.
   - Wenn nur das Kennwort für die Anmeldung erforderlich ist, fahren Sie mit Schritt 4 fort.
   - Wenn Anwendername und Kennwort für die Anmeldung erforderlich ist, dann lesen Sie aus der Telnet-Sitzung, bis eine der folgenden Aktionen eintritt:
     - Es ist eine Übereinstimmung der Muster vorhanden, die in der *Textaufforderung für Anwenderanmeldung* angegebenen sind. (Fahren Sie mit Schritt 3 fort.)
     - "Zeit für Warten auf Aufforderungen" ist abgelaufen. Wenn diese Zeit abläuft, schlägt der Operator fehl.
3. Senden Sie den Anwendernamen.
4. Lesen Sie aus der SSH-Sitzung, bis eine der folgenden Aktionen eintritt:
   - Es ist eine Übereinstimmung dem Muster vorhanden, das in "Textaufforderung für Anwenderkennwort" angegeben ist (fahren Sie mit Schritt 5 fort).
   - "Zeit für Warten auf Aufforderungen" ist abgelaufen. Wenn dies auftritt, brechen Sie den Operator ab.
5. Senden Sie dann das Kennwort:
   a. Lesen Sie aus der SSH-Sitzung, bis eine der folgenden Aktionen eintritt:
      - Es ist eine Übereinstimmung der Muster vorhanden, die in der *Eingabeaufforderung für Anwender* angegebenen sind, und Sie fahren fort.
      - "Zeit für Warten auf Aufforderungen" läuft ab. Brechen Sie dann den Operator ab.
   b. Es wird bestimmt, ob der Operator auf Befehl als anderer Anwender ausführen festgelegt werden soll:
      - Wenn ja, fahren Sie mit Schritt 6 fort.
      - Wenn nein, fahren Sie mit Schritt 7 fort.

6. Senden Sie den *Befehl für Anwenderwechsel*, und nehmen Sie Folgendes vor:
   a. Lesen Sie aus der SSH-Sitzung, bis eine der folgenden Aktionen eintritt:
      - Es ist eine Übereinstimmung der Muster vorhanden, die in der *Textaufforderung für Kennwort bei Anwenderwechsel* angegebenen sind.
      - Es ist eine Übereinstimmung der Muster vorhanden, die in der *Eingabeaufforderung für Anwenderwechsel* angegebenen sind.
      - "Zeit für Warten auf Aufforderungen" ist abgelaufen.
   b. Führen Sie basierend auf dem Ergebnis eine der folgenden Aktionen aus:
      - Wenn Sie das Muster für *Textaufforderung für Kennwort bei Anwenderwechsel* abgleichen (Kennwort erforderlich, um Anwender zu wechseln), senden Sie das *Kennwort für Anwenderwechsel,* und lesen Sie aus der Telnet-Sitzung, bis eine der folgenden Aktionen eintritt:

        - Es ist eine Übereinstimmung der Muster vorhanden, die in der Eingabeaufforderung für Anwenderwechsel angegebenen sind. Fahren Sie mit Schritt 7 fort.

        - "Zeit für Warten auf Aufforderungen" läuft ab und der Operator schlägt fehl.
      - Wenn Sie das Muster für *Eingabeaufforderung für Anwenderwechsel* abstimmen (kein Kennwort erforderlich, um Anwender zu wechseln), fahren Sie mit Schritt 7 fort.
      - Wenn "Zeit für Warten auf Aufforderungen" abläuft, brechen Sie den Operator ab.
7. Der Operator "Telnet-Befehl ausführen" hat sich angemeldet und gegebenenfalls den Anwender gewechselt. Der Operator ist nun bereit, die Befehle auszuführen.
8. Führen Sie die Befehle in einer Schleife aus, senden Sie gleichzeitig einen Befehl, und lesen Sie aus der Telnet-Sitzung, bis eine der folgenden Aktionen eintritt:
   - Es ist eine Übereinstimmung der Muster vorhanden, die in der Eingabeaufforderung für Anwender angegeben sind (wenn nicht zu einem anderen Anwender gewechselt wurde). Wiederholen Sie Schritt 8, und senden Sie den nächsten Befehl.
   - Es ist eine Übereinstimmung der Muster vorhanden, die in der Eingabeaufforderung für Anwenderwechsel angegeben sind (wenn zu einem anderen Anwender gewechselt wurde). Wiederholen Sie Schritt 8, und senden Sie den nächsten Befehl.
   - Der Wert in "Zeit für Warten auf Aufforderungen" läuft ab, ohne dass eine "Eingabeaufforderung für Anwender" oder "Eingabeaufforderung für Anwenderwechsel" (je nachdem, was zutreffend ist) vorhanden ist. Brechen Sie den Operator ab.

Operator "Telnet-Skript ausführen"

Der Operator "Telnet-Skript ausführen" verwendet den von Ihnen angegebenen Remote-Host und den Port, um Folgendes durchzuführen:

- Stellen Sie eine Telnet-Verbindung zum Remote-Host her.
- Ein "conn"-Objekt wird erstellt.

  Hinweis: Wenn Sie ein Skript angeben, entweder BeanShell oder JavaScript, wird das "conn"-Objekt im Bereich dieses Skripts verfügbar gemacht.

  Sie können die öffentlichen Methoden des "conn"-Objekts im Skript optimal nutzen. Die öffentlichen Methoden werden verwendet, um eine Authentifizierung in der Telnet-Sitzung durchzuführen und Betriebsabläufe zu automatisieren, die auf einem Telnet-Pseudoterminal ausgeführt wurden. Beispiele für automatisierte Betriebsabläufe sind:

  - Senden von Befehlen zum Remote-Telnet-Host.
  - Warten auf die Aufforderung warten, nachdem alle Befehle gesendet wurde.
  - Abrufen der Ausgabe von jedem Befehl.

Im Gegensatz zum Operator "SSH-Skript ausführen" authentifiziert der Operator "Telnet-Skript ausführen" keine Anwender-ID in der Telnet-Verbindung. Stattdessen müssen Sie die Methoden des "conn"-Objekts optimal nutzen, um in der Telnet-Verbindung am Anfang Ihres Skripts eine Authentifizierung durchzuführen.

Im Gegensatz zum Operator "Telnet-Befehl ausführen", bietet Ihnen der Operator "Telnet-Skript ausführen" folgende Möglichkeiten:

- Ermöglicht es Ihnen, anzugeben, welche Befehlsausgabe im Datensatz des Operators angezeigt oder gespeichert werden soll. Sie rufen folgende Methoden in dieser Sequenz auf:

  1. ‘conn.sendLine()’
  2. ‘conn.waitFor()’
  3. ‘conn.getLastOutput()’
- Ermöglicht es Ihnen, einen anderen regulären Ausdruck anzugeben, um nach Ausführung aller Befehle mit der Aufforderung übereinzustimmen.
- Ermöglicht es Ihnen, eine andere Zeit festzulegen, die nach Ausführung aller Befehle *auf die Eingabeaufforderung gewartet* werden soll.

Eingabeparameter

Nachfolgend sind die Eingabeparameter für den Operator "Telnet-Skript ausführen" aufgeführt.

Skript

Name des Remote-Host

Der Hostname oder die IP des Computers, zu der eine Verbindung hergestellt werden soll.

Skripttyp:

Der Typ des Skripts, das im Feld "Inline-Skript" angegeben ist. Wählen Sie zwischen BeanShell-Skript (.bsh) und JavaScript (.js) aus. Wenn dieses Feld leer gelassen wird, verwendet der Operator standardmäßig ".bsh".

Inline-Skript?

Dieser Operator stellt zwei Methoden bereit, um das Skript anzugeben: Inline oder als Ausdruck. Aktivieren Sie dieses Kontrollkästchen, um das Skript als Inline-Skript anzugeben.

Inline-Skript

Das Skript, das in BeanShell oder JavaScript geschrieben ist, verwendet das conn-Objekt und seine API folgendermaßen:

- Ein Befehl wird an den Remote-Host gesendet
- Es wird gewartet, bis der Befehl beendet wird
- Die Ausgabe des letzten Befehls wird abgerufen

Die APIs, die das conn-Objekt anzeigt, werden in Operator "Telnet-Skript ausführen" - Inline-Skript APIs (siehe Seite 144) ausführlich behandelt.

Skript als Ausdruck

Stellt das Skript als Ausdruck bereit. Informationen zum Skript selbst finden Sie im Feld "Inline-Skript".

Parameter

Die CA Process Automation-Parameter, die an das Skript übergeben werden sollen. Nur einfache CA Process Automation-Parametertypen können folgendermaßen an das Skript übergeben werden:

- PAM-Boolean wird als Boolean-Objekt übergeben.
- PAM-Date wird als Date-Objekt übergeben.

- PAM-Double wird als Double-Objekt übergeben.
- PAM-Integer wird als Integer-Objekt übergeben.
- PAM-Long wird als Long-Objekt übergeben.
- PAM-String wird als String-Objekt übergeben.
- PAM-Object Reference wird als Object Reference-Objekt übergeben.

Komplexe CA Process Automation-Parametertypen (indizierte Typen, ValueMaps usw.) können nicht an das Skript übergeben werden.

Das Skript kann auf diese Objekte über das args-Array der Objekte zugreifen, wobei args[0] dem ersten Parameter und args[1] dem zweiten Parametern in der Liste entspricht usw.

Ausgabevariablen-Namen

Die Namen der im Skript erstellten Variablen, die im Datensatz des Operators gespeichert werden sollen, nachdem die Ausführung des Skripts abgeschlossen wurde.

Die Variablen müssen im Bereich des Skripts definiert sein, damit sie nach der Ausführung sichtbar sind im Datensatz des Operators gespeichert werden können.

Die Ausgabevariablen werden folgendermaßen gespeichert:

- Boolean-Objekt wird als "PAM-Boolean" gespeichert
- Date-Objekt wird als "PAM-Date" gespeichert
- Integer-Objekt wird als "PAM-Integer" gespeichert
- Number-Objekt wird als "PAM-Long"- oder "PAM Double"-Objekt gespeichert
- String-Objekt wird als "PAM-String" gespeichert
- Character-Objekt wird als "PAM-String" gespeichert
- Ein Array der Objekte wird als indizierter PAM-Typ gespeichert, wobei der PAM-Typ durch den Typ des ersten Objekts im Array der Objekte definiert wird.
- "Undefined" wird als PAM-String mit dem Wert "undefined" gespeichert.

Operator "Telnet-Skript ausführen" - Inline-Skript APIs

Das Produkt verwendet für das Feld "Inline-Skript" des Operators "Telnet-Skript ausführen" ein Skript, das in BeanShell oder JavaScript geschrieben ist. Das Skript verwendet das Objekt "conn", das die folgenden APIs darstellt:

```
void send (String str, boolean log) throws Exception
```

Diese Methode sendet Daten an den Remote-Host.

String str

Definiert die Daten, die an den Remote-Host gesendet werden sollen.

boolean log

Stellt die Daten, die an den Remote-Host gesendet werden, in den CA Process Automation-Protokollen dar bzw. blendet sie aus.

Folgen Sie für Debugging-Zwecke der Interaktion zwischen dem Operator und dem Remote-Host. Legen Sie in der Datei <*Installationsverzeichnis*>/server/conf/log4j.xml den folgenden Code fest:

```
<category name="com.company.c2o.servicegroup.netutils">


   <priority value="DEBUG" />
</category>
```

Legen Sie außerdem fest, dass die CA Process Automation-Protokolldatei (c2o.log) DEBUG-Anweisungen in "log4j.xml" akzeptiert.

Wenn Sie die Debug-Ebene festgelegt haben, beginnt der Operator "Befehlsausführung", auf der DEBUG-Ebene in die CA Process Automation-Protokolldatei (c2o.log) zu schreiben. Die CA Process Automation-Protokolle enthalten sämtliche Daten, die über "send" oder "sendLine" an den Remote-Host gesendet wurden.

Werte:

Wahr: Protokolliert den Wert "String str", wenn auf Debug-Ebene protokolliert wird.

Falsch: Protokolliert den Wert "String str" nicht.

Hinweis: "c2o.log" ist die CA Process Automation-Protokolldatei, *nicht* das Prozessprotokoll. Die Operatoren schreiben keine Meldungen in das Prozessprotokoll.

Diese Methode hat keine Rückgabewerte. Das Produkt generiert eine Ausnahme, wenn die API keine Daten im Remote-Host schreiben kann.

`public void sendLine (String str, boolean log) throws Exception`

Diese Methode hängt ein Zeichen für neue Zeile an die Daten an und sendet sie an den Remote-Host. Um zu erzwingen, dass der Remote-Host mit der Ausführung des Befehls beginnt, der im Parameter gesendet wird, verwenden Sie diese Methode.

Diese Methode hat die gleichen Parameter und Werte wie "void send (String str, boolean log) throws Exception".

`void send (String str) throws Exception`

Diese Methode entspricht "Send" (String str, "Wahr").

`void sendLine (String str) throws Exception`

Diese Methode entspricht "SendLine" (String str, "Wahr").

`public boolean waitFor(String pattern, int timeout) throws Exception`

Diese Methode liest die Ausgabe vom Remote-Host und hält an, wenn eines der folgenden Ergebnisse auftritt:

- Die Ausgabe stimmt mit dem Muster überein, das in den Parametern angegeben ist.
- Das Zeitlimit-Intervall läuft ab.

Das Produkt speichert die Ausgabe aus dem Remote-Host der einzelnen Aufrufe der Methode "waitFor" in einem Puffer, auf den über die Methode "getLastOutput()" zugegriffen werden kann. Jeder Aufruf von "waitFor" überschreibt den Pufferinhalt des vorherigen Aufrufs.

Der nächste Aufruf von "waitFor" liest die Ausgabe ab dem Punkt, an dem der vorhergehende waitFor-Aufruf zu lesen aufgehört hat. Dies ist relevant, wenn Sie die Methode mit "getLastOutput.()" verwenden. Wenn ein waitFor-Aufruf beispielsweise nicht mit der ganzen Befehlsausgabe übereinstimmt, kann der nächste Aufruf die verbleibende Ausgabe des früheren Befehls enthalten.

Wichtig! Um zu vermeiden, dass die Ausgabe des früheren Befehls mit der aktuellen Befehlsausgabe vermischt wird, rufen Sie nach jedem sendLine-Aufruf "waitFor" auf.

Die Operatoren "Telnet-Skript ausführen" und "SSH-Skript ausführen" verwenden verschiedene Mechanismen, um Daten von einem Remote-Host zu lesen:

- Bei Telnet liest "waitFor" die Daten direkt vom Host und gleicht sie während des Lesevorgangs ab.
- Bei SSH ruft "waitFor" die bisher gelesenen Daten aus einem Puffer ab und gleicht sie mit dem Muster ab.
- Wenn Sie "waitFor" nach jedem sendLine-Aufruf aufrufen, ist das Verhalten von SSH- und Telnet-Operatoren identisch.

Zu den Parametern gehören:

String pattern

Definiert einen regulären Ausdruck, mit dem die Daten, die das Produkt vom Remote-Host liest, abgeglichen werden sollen. Normalerweise steht dieses Muster für alle Daten bis zur nächsten Aufforderung (zum Beispiel ".*[$]"). Um die Befehlsausgabedaten einschließlich neuer Zeilen bis zur nächsten Aufforderung abgleichen (und abrufen) zu können, beginnen Sie das Muster mit ".*.".

Hinweis: Die Methode gleicht alle gelesenen Daten (während dieses Aufrufs von "waitFor") mit dem Muster ab. Die Methode entspricht nicht dem Muster als untergeordnete Zeichenfolge der gelesenen Daten. Ein Punkt (.) kann außerdem mit einer Zeilenendmarke übereinstimmen (er kann verwendet werden, um mit Antwortdaten, die sich über mehrere Zeilen erstrecken, übereinzustimmen).

Int timeout

Definiert das Intervall (in Sekunden), während dessen Daten vom Remote-Host gelesen und mit dem Muster abgeglichen werden sollen.

Rückgaben sind Boolesch:

Wahr

Die Daten, die bis zur Zeitüberschreitung gelesen wurden, stimmen mit dem Muster überein.

False

Die Daten, die bis zur Zeitüberschreitung gelesen wurden, stimmen nicht mit dem Muster überein.

Hinweis: Die Methode gibt einen Wert zurück, sobald eine Übereinstimmung mit dem Muster gefunden wird oder die Zeitüberschreitung erreicht ist.

In den folgenden Fällen generiert das Produkt eine Ausnahme:

- Der Parameter "Int timeout" ist kleiner oder gleich Null.
- Der Parameter "String pattern" ist ungültig oder leer.
- Das Produkt kann keine Daten aus der Telnet-Sitzung lesen.
- Das Muster enthält ungültige Syntax.
- Beim Abgleichen des Musters mit den empfangenen Daten tritt ein Fehler auf.

```
public String getLastOutput()
```

Diese Methode gibt den Inhalt des Puffers zurück, an dem die Daten beim letzten Aufruf der Methode "waitFor" gespeichert und vom Remote-Host gelesen wurden. Diese Daten stimmen nicht immer mit dem waitFor-Muster überein. Der Puffer speichert die Werte, die durch den letzten waitFor-Aufruf gelesen wurden, unabhängig davon, ob "waitFor" "true" oder "false" zurückgegeben hat.

Diese Methode hat keine Parameter.

Diese Methode generiert keine Ausnahmen.

Informationen für Remote-Anmeldung

Pseudoterminaltyp

Der Pseudoterminaltyp, der bei Telnet-Verbindungen angefordert wird. Dieses Feld überschreibt den Wert, der auf der Modulebene angegeben ist. Wenn das Feld leer gelassen wird, verwendet der Operator den Standardwert, der auf der Modulebene festgelegt ist. Wenn dieser Wert leer ist, wird der Operator standardmäßig auf VT100 festgelegt.

- VT100 funktioniert normalerweise mit den meisten Computern (besonders Linux-basiert).
- VT400 funktioniert normalerweise mit Windows-basierten Computern. VT400 ist für Windows-Plattformen erforderlich, besonders wenn die Ausgabe, die aus dem Telnet-Server (Befehlsausgabe) abgerufen wird, Steuerzeichen anstatt Leerzeichen enthält. Zum Beispiel wenn in der Ausgabe "[19;1H" statt Leerzeichen enthalten ist. VT400 interpretiert die Leerzeichen für Windows richtig.

Andere Terminaltypen können verwendet werden. Stellen Sie sicher, dass Sie sie testen, bevor Sie den Operator in die Produktion verschieben. Einige Pseudoterminaltypen sind:

- dumb
- xterm
- vt220
- vt320
- gogrid

Überprüfen Sie die Installation und Konfiguration Ihres Telnet-Servers für den unterstützten Pseudoterminal.

Der Pseudoterminaltyp steuert, wie Leerzeichen in der Befehlsausgabe angezeigt werden. Sie sollten diesen Operator gegen die Pseudoterminals testen, die vom Telnet-Server unterstützt werden, um einen geeigneten Pseudoterminal zu finden, der die Leerzeichen entsprechend zurückgibt. Wenn die Leerzeichen nicht entsprechend zurückgegeben werden und kein Pseudoterminal, der diesen Issue beheben könnte, verfügbar ist, gehen Sie folgendermaßen vor:

- Ändern Sie die Eingabe des Operators, um die Beschränkung dieses Telnet-Servers unterzubringen.
- Verwenden Sie JavaScript, um die Ausgabe der Befehle zu verfeinern bzw. zu extrahieren.

Wenn Sie einen Pseudoterminal anfordern, der nicht unterstützt wird, geben einige Telnet-Server einen Fehler zurück, während andere Server den angeforderten Pseudoterminaltyp ignorieren und andere Pseudoterminaltypen verwenden. Überprüfen Sie die Protokolle des Telnet-Servers für den Pseudoterminal, die verwendet werden, wenn der Operator ausgeführt wird.

Remote-Port

Der Port zum Verbinden mit dem Remote-Host. Dieses Feld überschreibt den Wert, der auf der Modulebene angegeben ist. Wenn dieses Feld leer gelassen wird, verwendet der Operator den Standardwert, der auf der Modulebene festgelegt ist. Wenn dieser standardmäßige Modulwert leer ist, verwendet der Operator den Wert: 23.

Zeitlimit für Verbindung (Sek.)

Das Verbindungszeitlimit, nach dessen Ablauf der Verbindungsversuch abgebrochen wird. Dieses Feld überschreibt den Wert, der auf der Modulebene angegeben ist. Wenn dieses Feld leer gelassen wird, verwendet der Operator den Standardwert, der auf der Modulebene festgelegt ist. Wenn dieser standardmäßige Modulwert leer ist, verwendet der Operator den Wert: 20.

Ausgabeparameter

Alle Variablen in der Liste "Ausgabevariablen-Namen" werden mit dem entsprechenden CA Process Automation-Typ erstellt.

Wenn ein Variablenname in "Ausgabevariablen-Namen" nicht im Skript vorhanden ist, erstellt der Operator "Telnet-Skript ausführen" die entsprechende Variable als leere Zeichenfolge.

Der BeanShell-Interpreter stellt eine robuste Umgebung für das Skript bereit. Wenn das BeanShell-Skript beispielsweise eine Ausnahme ausgibt oder einen Fehler enthält, können die Variablen, die vor dem Auftreten des Fehlers definiert und im Skript initialisiert wurden, mit ihren Werten abgerufen werden. Wenn das JavaScript-Skript eine Ausnahme auslöst oder einen Fehler enthält, können alternativ mit dem JavaScript-Interpreter keine Variablen mit ihrem eigenen Wert abgerufen werden.

Zu den Ausgabeparametern gehören:

TelnetHost

inLineScriptLanguage

inlineScriptType

inLineScript

scriptExpression

parameters

outputVariables

TelnetPseudoTerminalType

TelnetPort

TelnetHostConnectTimeout

Operator-Ports

Erfolg

Der Operator wurde erfolgreich abgeschlossen.

Fehlgeschlagen

Der Operator schlägt aus folgenden Gründen fehl:

- Der Anwender gibt ungültige Eingabe an. Zum Beispiel: Leeres Inline-Skript, leerer Remote-Host, negativer Port usw.
- Der Anwender gibt einen anderen Inline-Skript-Typ als ".bsh" und ".js" an.
- Es kann keine Verbindung zum Remote-Host hergestellt werden. Überprüfen Sie den im angegebenen Operator Remote-Host und Port.
- Der Anwender gibt einen unbekannten Host im Remote-Host an.
- Der Anwender gibt einen komplexen Datentyp in der Liste von Parametern an. Komplexe Werttypen (indizierte Typen, ValueMaps usw.) können nicht an das Skript übergeben werden. Verwenden Sie einfache Werttypen wie "Doppelt", "Ganzzahl", "Lang", "Zeichenfolge", "Datum" und "Boolesch".
- Beim Ausführen eines ".bsh"- oder ".js"-Skripts:
  - Beim Auswerten des Skripts tritt ein Analyse- oder Syntaxfehler auf.
  - Das Skript gibt eine Ausnahme aus.
  - Beim Ausführen des Skripts tritt ein Fehler auf.
  - Fehler beim Abrufen einer Variablen aus dem Skriptbereich (am Ende der Ausführung).

Anwenderspezifische Ports

Wenn während des Prozessdesigns durch den Anwender festgelegt.

Beispiel

Das folgende Verfahren ist ein Beispiel für die Verwendung des Operators "Telnet-Skript ausführen".

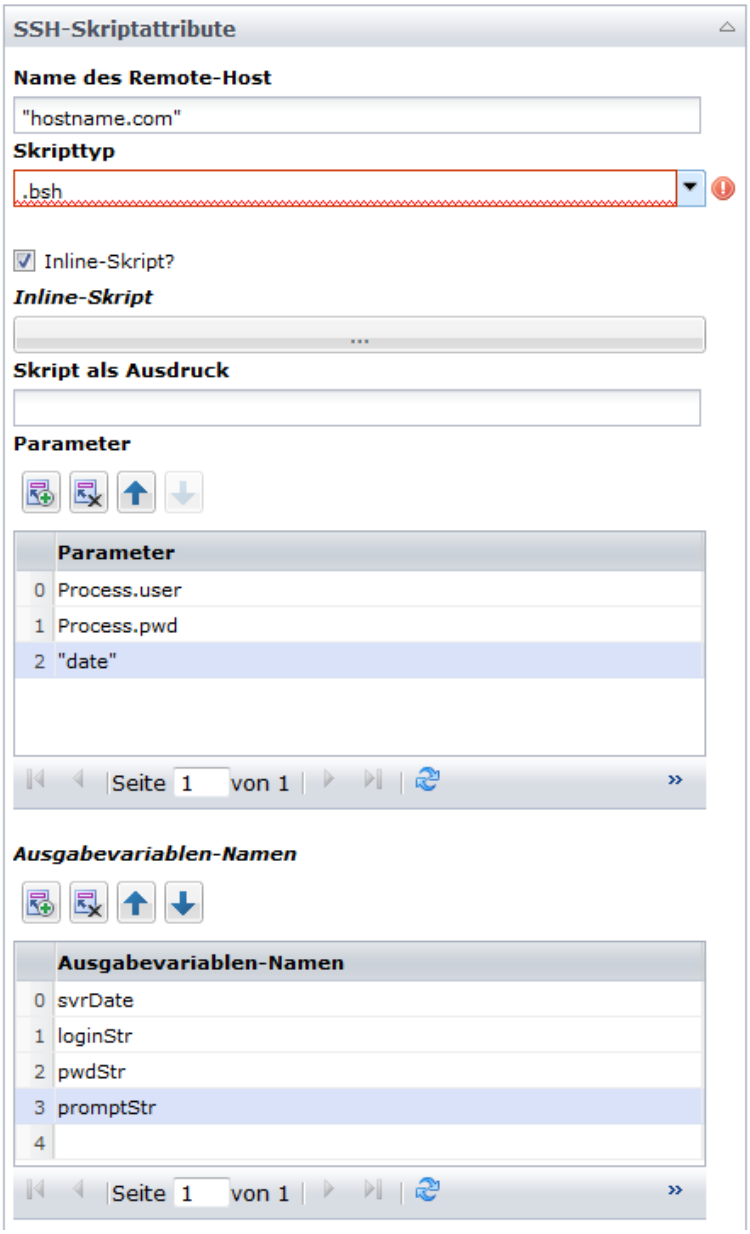
1. Der Operator "Telnet-Skript ausführen" liest die von Ihnen angegebenen Anmeldeinformationen. Der Operator verwendet diese Anmeldeinformationen, um eine Verbindung mit dem Remote-Telnet-Host herzustellen und um das "conn"-Objekt zu erstellen. Das folgende Beispiel für Informationen für Remote-Anmeldung zeigt typische Eingaben:

2. Im Auswahlmenü "Skript" geben Sie Folgendes an:

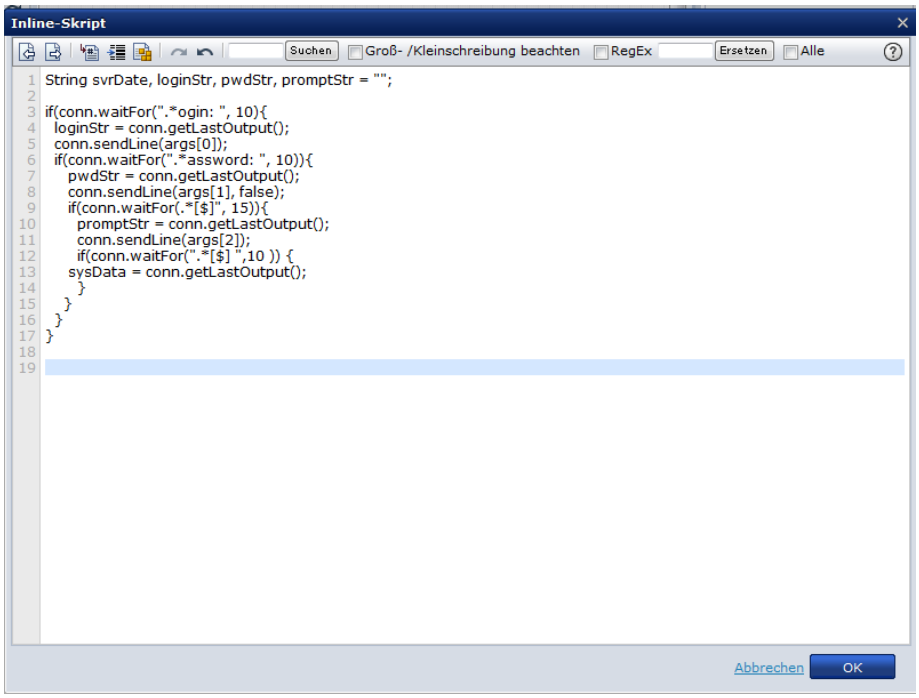
   - Den Namen des Remote-Host.
   - Die Parameter, die an das Skript übergeben werden sollen. Im folgenden Beispiel der Anwender, das Kennwort und eine Zeichenfolge, deren Wert das Wort "date" ist.
   - Der Namen der Ausgabevariable, die Sie im Skript erstellt haben (BeanShell oder JavaScript) und das im Datensatz des Operators gespeichert werden sollen, nachdem die Ausführung abgeschlossen wurde. Wenn Sie nicht möchten, dass am Ende der Ausführung eine Variable in den Datensatz des Operators gespeichert wird, dann müssen Sie dies hier nicht angeben.

     Hinweis: Die Erstellung von Datensatzvariablen direkt aus dem Skript wird derzeit nicht unterstützt. Das Feld "*Ausgabevariablen-Namen*" spielt in diesem Fall die Rolle "C2OSVD".

   In diesem Beispiel werden am Ende der Ausführung folgende Ausgabevariablen-Namen im Datensatz des Operators gespeichert: "svrDate", "loginStr", "pwdStr" und "promptStr".


SSH-Skriptattribute
Name des Remote-Host
"hostname.com"
Skripttyp
.bsh
Inline-Skript?
Inline-Skript
Skript als Ausdruck
Parameter
Parameter
0 Process.user
1 Process.pwd
2 "date"
Seite 1 von 1
Ausgabevariablen-Namen
Ausgabevariablen-Namen
0 svrDate
1 loginStr
2 pwdStr
3 promptStr
4
Seite 1 von 1

3. Im Inline-Skript können Sie das von CA Process Automation erstellte "conn"-Objekt optimal nutzen. So können Sie das "conn"-Objekt, das im folgendem BeanShell-Skriptbeispiel angezeigt wird, optimal nutzen:



   a. Erstellen Sie die Variablen "svrDate", "loginStr", "pwdStr" und "promptStr" so, dass sie im Skriptbereich sichtbar sind, damit sie am Ende der Ausführung im Datensatz des Operators gespeichert werden können.

   b. Verwenden Sie *conn.waitFor()*, um bis zu 10 Sekunden auf die Anmeldeaufforderung "*.*nmeldung*" (Reg Ex) zu warten.

   c. Wenn die Anmeldeaufforderung innerhalb von 10 Sekunden gefunden wird, dann verwenden Sie *conn.getLastOutput()*, um die Anmeldeaufforderung zu speichern, die mit der Variablen "loginStr" übereinstimmt.

   d. Verwenden Sie *conn.sendLine(),* um den Anwendernamen zu senden, der als erstes Objekt in der Liste von Parametern übergeben wurde: *args[0]*.

   e. Verwenden Sie *conn.waitFor()*, um bis zu 10 Sekunden auf die Kennwortaufforderung "*.*ennwort*" (Reg Ex) zu warten.

   f. Wenn die Kennwortaufforderung innerhalb von 10 Sekunden gefunden wird, dann verwenden Sie *conn.getLastOutput()*, um die Kennwortaufforderung zu speichern, die mit der Variablen "pwdStr" übereinstimmt.

   g. Verwenden Sie *conn.sendLine(),* um das Kennwort zu senden, das als zweites Objekt in der Liste von Parametern übergeben wurde: *args[1]*.

h. Verwenden Sie *conn.waitFor()*, um bis zu 15 Sekunden auf die Anwendereingabeaufforderung "*.*[$]* " (Reg Ex) zu warten.

i. Wenn die Anwendereingabeaufforderung innerhalb von 15 Sekunden gefunden wird, dann verwenden Sie *conn.getLastOutput()*, um die Anwendereingabeaufforderung zu speichern, die mit der Variablen "promptStr" übereinstimmt.

j. Verwenden Sie *conn.sendLine(),* um den Befehl zu senden, der als drittes Objekt in der Liste von Parametern übergeben wurde: *args[2]*.

k. Verwenden Sie *conn.waitFor()*, um bis zu 10 Sekunden auf die Anwendereingabeaufforderung "*.*[$]* " (Reg Ex) zu warten.

l. Wenn die Anwendereingabeaufforderung innerhalb von 10 Sekunden gefunden wird, dann verwenden Sie *conn.getLastOutput()*, um die übereinstimmende Anwendereingabeaufforderung mit der Ausgabe des Befehls in der Variablen "svrDate" zu speichern.

m. Am Ende der Ausführung speichert der Operator "Telnet-Skript ausführen" die Ausgabevariablen im Datensatz des Operators als CA Process Automation-Objekte.

   Die übereinstimmende Aufforderung wird in der Ausgabe angezeigt, die von *conn.getLastOutput()* zurückgegeben wird.

Kapitel 5: Datenbanken

Datenbanken-Operatoren unterstützen JDBC-Treiber vom Typ 2, um mit einer Datenbank zu kommunizieren. Datenbank-Operatoren können unterschiedliche Datenbankvorgänge ausführen, beispielsweise Abfragen auf der Datenbank. Sie unterstützen jedoch keine Verwaltungsvorgänge, wie etwa das Anhalten eines Datenbankservers, Sicherungen/Wiederherstellungen usw.

Die Verbindungszeichenfolge unterscheidet sich je nach Servertyp.

Die Kombination von TNS-Name und Thin-Treibertyp wird nur in Oracle 12.*x* Version 12.x und höher unterstützt. CA Process Automation validiert weder jede Kombination noch die auf Operatorebene bereitgestellten Serveranmeldedaten.

Hinweis: Sie können die Datenbanken-Operatoren mit einem anderen RDBMS (Relational Database Management System) als das von den CA Process Automation-Datenbanken verwenden. Wenn CA Process Automation beispielsweise mithilfe von Microsoft SQL installiert wurde, können Sie die Datenbanken-Operatoren mit Oracle verwenden. Allerdings muss zunächst die entsprechende Datenbanktreiberdatei korrekt in Ihrer CA Process Automation-Installation bereitgestellt werden. Weitere Informationen finden Sie im *Installationshandbuch*.

Oracle-Parameter

Einstellungen erben

Wenn dies aktiviert ist, spiegeln die gezeigten Werte die aktuellen Werte der Domäne wider. Zur Laufzeit werden die Werte aus der Umgebung ausgewählt (sofern definiert).

Treibertyp

Akzeptiert eine der folgenden Optionen:

Thin

Der Thin-Treiber ist eine reine Java-Implementierung des Netzwerkprotokolls von Oracle (Net8). Da sie selbständig ist, kann sie auf Rechnern mit oder ohne Oracle verwendet werden oder mit Anwendungsklassen in einem Applet verteilt werden.

OCI

Der "OCI" (Typ 2)-Treiber besteht aus Java-Wrapper zu Oracle Call Interface (OCI)-Bibliotheken auf unterer Ebene, die von Hilfsprogrammen wie SQL*Plus verwendet werden, um auf den Datenbankserver zuzugreifen. Der OCI-Treiber bietet potenziell bessere Leistung als der Thin-Treiber. Allerdings müssen die OCI-Bibliotheken auf dem lokalen Rechner installiert sein.

KPRB

Der "KPRB"-Treiber wird für Java-gespeicherte Vorgänge und Datenbank-JSPs verwendet.

Treiber

Gibt den Oracle-JDBC-Treiber an.

Serverhost

Gibt den Host an, auf dem die Oracle-Datenbank ausgeführt wird.

UserName

Gibt den Standardanwender der Oracle-Datenbank an.

Kennwort

Gibt das Kennwort für den Oracle-Standardanwender an.

ServiceID

Gibt die Oracle-Service-ID an.

TNS Name

"Oracle-TNS-Name" übersetzt ein lokales Datenbankalias in alle Konnektivitätsinformationen, die für die Verbindung zur Datenbank erforderlich sind. Dies schließt IP-Adresse, Port, Datenbankdienst-ID oder Servicenamen usw. ein. Diese Informationen werden in der Datei "tnsnames.ora" im Oracle-Verzeichnis gespeichert.

Maximale Spaltenanzahl

Gibt die maximale Anzahl der abzurufenden Zeilen an.

Client-Verschlüsselung

Oracle unterstützt mehrere Datenverschlüsselungen für den Client (RC4_40, RC4_56, RC4_128, RC4_256, DES40C, DES56C, 3DES112, 3DES168, SSL, AES256, AES192 und AES128). Der Anwender sollte einen dieser Werte angeben. Diese Werte werden als Eigenschaften als Teil der Verbindung festgelegt. Die Verschlüsselungsstufen "RC4_128" und "RC4_256" gelten nur für private Ausgaben.

Client-Prüfsumme

Gibt die von Oracle unterstützten Prüfsummen an. Weitere Informationen finden Sie in der Oracle-Dokumentation.

MSSQL-Server-Parameter

Einstellungen übernehmen

Wenn dies aktiviert ist, spiegeln die gezeigten Werte die aktuellen Werte der Domäne wider. Zur Laufzeit werden die Werte aus der Umgebung ausgewählt (sofern definiert).

Standardmäßiger Treiber

Gibt den MSSQL-Standardtreiber an.

Standardmäßiger Serverhost

Gibt den Host an, auf dem die MSSQL-Datenbank ausgeführt wird.

Standardmäßiger Serverport

Gibt den Standardport des MSSQL-Datenbankservers an.

Standardmäßiger Anwendername

Gibt den Standardanwender der MSSQL-Datenbank an.

Standardmäßiges Kennwort

Gibt das Kennwort für den MSSQL-Standardanwender an.

Standardmäßige maximale Zeilenanzahl

Gibt die größtmögliche Anzahl von abzurufenden Zeilen an.

Standardmäßiger Datenbankname

Gibt den MSSQL-Datenbanknamen an.

Standardmäßiger Instanzenname

Gibt den MSSQL-Instanznamen an.

Parameter von MySQL

Einstellungen übernehmen

Wenn dies aktiviert ist, spiegeln die gezeigten Werte die aktuellen Werte der Domäne wider. Zur Laufzeit werden die Werte aus der Umgebung ausgewählt (sofern definiert).

Standardmäßiger Treiber

Gibt den MySQL-Standardtreiber an.

Standardmäßiger Serverhost

Gibt den Host an, auf dem die MySQL-Datenbank ausgeführt wird.

Standardmäßiger Serverport

Gibt den Standardport des MySQL-Datenbankservers an.

Standardmäßiger Anwendername

Gibt den Standardanwender der MySQL-Datenbank an.

Standardmäßiges Kennwort

Gibt das Kennwort für den MySQL-Standardanwender an.

Standardmäßige maximale Zeilenanzahl

Gibt die größtmögliche Anzahl von abzurufenden Zeilen an.

Standardmäßiger Datenbankname

Gibt den Standardnamen der MySQL-Datenbank an.

Sybase-Parameter

Einstellungen erben

Wenn dies aktiviert ist, spiegeln die gezeigten Werte die aktuellen Werte der Domäne wider. Zur Laufzeit werden die Werte aus der Umgebung ausgewählt (sofern definiert).

Standardmäßiger Servertyp

Gibt einen der folgenden Sybase-Servertypen an:

- Adaptive Server Anywhere (ASA) (Standardwert)
- Adaptive Server Enterprise (ASE)

Standardmäßiges Verbindungsprotokoll

Gibt das Standardverbindungsprotokoll an. Der Standardwert lautet Tds.

Standardmäßiger Treiber

Gibt den Sybase-Standardtreiber an. Der Standardwert lautet com.sybase.jdbc2.jdbc.SybDriver.

Standardmäßiger Serverhost

Gibt den Host an, auf dem Sybase ausgeführt wird.

Standardmäßiger Serverport

Gibt den Standardport des Sybase-Servers an.

Standardmäßiger Anwendername

Gibt den Sybase-Standardanwendernamen an.

Standardmäßiges Kennwort

Gibt das Kennwort für den Sybase-Anwender an.

Standardmäßige maximale Zeilenanzahl

Gibt die größtmögliche Anzahl von abzurufenden Zeilen an. Wenn leer, lautet der Standardwert 10 Zeilen.

Standardgröße des Zwischenspeicherpuffers

Die Puffergröße des Sybase-Zwischenspeichers ist die Größe des vom Treiber für die Zwischenspeicherung von nicht vertraulichen Ergebnissatzdaten verwendeten Speichers. Gültige Werte sind:

-1 = Alle Daten werden zwischengespeichert.

0 = Alle Daten bis zu 2 GB werden zwischengespeichert.

X = Müssen positiv sein. Dies ist die Puffergröße (muss ein Mehrfaches von 2 sein). Dieser Wert wird in KB angegeben.

Nachdem das Limit erreicht wurde (sofern vorhanden), werden die Ergebnissatzdaten auf dem Datenträger abgespeichert.

Standardmäßige Umgehungslösung für Batch-Verarbeitung

Die Umgehungslösung für die Sybase-Batch-Verarbeitung ist eine der Folgenden:

- "Wahr" steht für den JDBC 3.0-kompatiblen Mechanismus.
- "Falsch" steht für den systemeigenen Batchmechanismus. Der Standard lautet "Falsch".

Eigenschaften auf Operatorebene

Nachfolgend werden Verbindungsparameter für die Datenbankoperatoren angegeben.

Anmeldeparameter für den Datenbankserver

Die Anmeldeparameter für den Datenbankserver konfigurieren Einstellungen, die erforderlich sind, um sich beim Datenbankserver anzumelden und um mit dem Datenbankserver zu kommunizieren.

Anwendername

Gibt den Datenbankanwendernamen an.

Kennwort

Gibt das Kennwort für den Datenbankanwender an.

Hinweise:

- Wenn als Datenbanktyp "Andere" ausgewählt wurde, geben Sie den Anwendernamen und das Kennwort ein, mit denen Sie sich an der Datenbank anmelden können (sofern erforderlich). Der Verbindungsassistent richtet eine URL ein, die unter den Operatoreigenschaften aufgefüllt wird.
- Wenn Sie die Windows-Authentifizierung verwenden möchten, geben Sie *keinen* Anwendernamen bzw. kein Kennwort an, wenn Sie einen Datenbankoperator konfigurieren. Weitere Informationen zur Konfiguration der Windows-Authentifizierung für die Datenbankoperatoren finden Sie im *Handbuch für Inhaltsadministratoren*.

Verbindungsassistent

Ein Assistent, mit dem Sie Verbindungseigenschaften angeben können. Sie können die Eigenschaften (siehe Seite 161) eingeben, die konfigurieren, wie der Operator mit der Datenbank eine Verbindung herstellt.

Verbindungs-URL

Stellt einen Universal Resource Locator (URL) dar, der einen bestimmten Datenbankserver-Typ (kompatibel mit dem lokalen JDBC-Treiber) und einen bestimmten Host angibt.

Hinweise:

- Wenn "Andere" als Datenbanktyp (im Verbindungsassistenten) ausgewählt wird, muss eine JDBC-URL in dieses Feld eingegeben werden.
- Wenn Sie die Windows-Authentifizierung verwenden möchten, hängen Sie Folgendes an die Verbindungs-URL an:

```
;integratedSecurity=true
```

Eigenschaften des Verbindungsassistenten

Datenbanktyp

Wählen Sie aus der Dropdown-Liste den Datenbanktyp:

- MySQL (Standard)
- Oracle
- SQLSERVER
- Sybase
- Informix
- Hypersonic
- Postgres
- DB2
- InterBase
- Ingres
- Sonstige

Datenbanktyp, Serverhost, Serverport, Datenbankname, Treibername und Verbindungs-URL werden immer im Verbindungsassistent angezeigt. Die verbleibenden Felder werden basierend auf der Auswahl des Datenbanktyps ein- oder ausgeblendet.

Wenn Sie beispielsweise "Oracle" als Datenbanktyp auswählen, werden alle mit Oracle zusammenhängenden Felder angezeigt und die übrigen Felder sind ausgeblendet.

Hinweis: Wenn Sie SQLSERVER als Datenbanktyp auswählen und Windows-Authentifizierung (integrierte Sicherheit) verwenden wollen, lassen Sie alle Felder außer "Verbindungs-URL" leer. Nehmen Sie in diesem Feld eine Eingabe vor, die dem folgenden Beispiel ähnlich ist:

"jdbc:sqlserver//ms-db-host:1433;DatabaseName=dbname;integratedSecurity=true
"

Wenn Sie "Host", "Port", "Datenbankname" oder "SQL Server-Instanzenname" definieren, erstellt der Operator die URL auf der Grundlage dieser Werte anstelle der Werte, die Sie im Feld "Verbindungs-URL" (mit "integratedSecurity" auf "true" gesetzt) konfiguriert haben.

Hinweis: Wenn als Datenbanktyp "Andere" ausgewählt wurde, geben Sie den Anwendernamen und das Kennwort, mit denen Sie sich an der Datenbank anmelden können (sofern erforderlich), unter "Anmeldeparameter für den Datenbankserver" ein.

Anderer Datenbanktyp

Wenn Ihre Datenbank nicht in der Dropdown-Liste "Datenbanktyp" aufgelistet wird, geben Sie sie hier ein.

Serverhost

Gibt den Host an, auf dem die Datenbank ausgeführt wird.

Serverport

Gibt den Datenbankserverport an.

Datenbankname

Gibt den Namen der Datenbank an.

Treibername

Gibt den Datenbanktreibernamen an (die Java-Klasse der Schnittstelle mit der Datenbank)

Verbindungs-URL

Legt eine Datenbank-URL als Universal Resource Locator (URL) fest, der einen bestimmten Datenbankserver-Typ (kompatibel mit dem lokalen JDBC-Treiber) und einen bestimmten Host angibt.

Dieses Feld wird aktualisiert, wenn Daten in den Verbindungsassistenten eingegeben werden.

Die folgenden Eigenschaften werden nur angezeigt, wenn sie für den angegebenen Datenbanktyp gelten.

Größe des Sybase-Zwischenspeicherpuffers

Verfügbar, wenn als Datenbanktyp "Sybase" ausgewählt wurde. Dieses Feld gibt den Speicherplatz an, der vom Treiber zur Zwischenspeicherung unempfindlicher Ergebnissatzdaten verwendet wird. Gültige Werte sind:

-1

Alle Daten werden zwischengespeichert.

0

Alle Daten bis zu 2 GB werden zwischengespeichert.

X

Dies ist die Puffergröße; sie muss positiv und eine Zweierpotenz sein. Dieser Wert wird in Kilobyte angegeben.

Umgehungslösung für Sybase-Batch-Verarbeitung

Verfügbar, wenn als Datenbanktyp "Sybase" ausgewählt wurde. Wählen Sie entweder "Wahr" oder "Falsch" aus.

- Der Standard lautet "Falsch" (systemeigener Batch-Mechanismus).
- "Wahr" steht für den JDBC 3.0-kompatiblen Mechanismus.

Sybase-Verbindungsprotokoll

Verfügbar, wenn als Datenbanktyp "Sybase" ausgewählt wurde. Gibt das Verbindungsprotokoll für Sybase an. Das Standardverbindungsprotokoll ist "Tds". Die Verbindungszeichenfolge unterscheidet sich je nach Servertyp.

Sybase-Servertyp

Verfügbar, wenn als Datenbanktyp "Sybase" ausgewählt wurde. Gibt die Sybase-Servertypen an. Wählen Sie eine der folgenden Optionen in der Dropdown-Liste aus:

- Adaptive Server Anywhere (ASA) (Standardeinstellung)
- Adaptive Server Enterprise (ASE)

Typ des Oracle-Treibers

Verfügbar, wenn als Datenbanktyp "Oracle" ausgewählt wurde. Gibt den Treibertyp für Oracle an. Wählen Sie eine der folgenden Optionen in der Dropdown-Liste aus:

thin

Der Thin-Treiber ist eine reine Java-Implementierung des Oracle-Netzwerkprotokolls (Net8). Da sie selbständig ist, kann sie auf Rechnern mit oder ohne Oracle verwendet werden oder mit Anwendungsklassen in einem Applet verteilt werden.

OCI

Der OCI-Treiber (Typ 2) besteht aus Java-Wrappern auf die Low-Level-Bibliotheken der Oracle-Aufrufschnittstelle (OCI), wie sie von Hilfsprogrammen wie SQL*Plus für den Zugriff auf den Datenbankserver verwendet werden. Der OCI-Treiber kann potenziell die Leistung über den Thin-Treiber verbessern, allerdings müssen dazu die OCI-Bibliotheken auf einem lokalen Rechner installiert werden.

KBRP

Der "KPRB"-Treiber wird für Java-gespeicherte Vorgänge und Datenbank-JSPs verwendet.

ID des Oracle-Services

Verfügbar, wenn als Datenbanktyp "Oracle" ausgewählt wurde. Ein Supportausdruck, der die Oracle-Service-ID angibt.

Oracle-TNS-Name

Verfügbar, wenn als Datenbanktyp "Oracle" ausgewählt wurde. Übersetzt ein lokales Datenbankalias in alle Konnektivitätsinformationen, die für die Verbindung zur Datenbank erforderlich sind. Dies schließt IP-Adresse, Port, Datenbankdienst-ID oder Servicenamen usw. ein. Diese Informationen werden in der Datei "tnsnames.ora" im Oracle-Verzeichnis gespeichert.

Oracle Client-Verschlüsselung

Verfügbar, wenn als Datenbanktyp "Oracle" ausgewählt wurde. Oracle unterstützt die folgenden Mehrfach-Datenverschlüsselungen für den Client:

RC4_40, RC4_56, RC4_128, RC4_256

DES40C, DES56C, 3DES112, 3DES168

SSL, AES256, AES192, AES128

Geben Sie einen dieser Werte an, die als Eigenschaften als Teil der Verbindung festgelegt werden. Die Verschlüsselungsstufen "RC4_128" und "RC4_256" gelten nur für private Ausgaben.

Prüfsumme des Oracle-Client

Verfügbar, wenn als Datenbanktyp "Oracle" ausgewählt wurde. Gibt den Prüfsummenwert des Oracle-Clients an (eine von der Datenbank berechnete Zahl von allen in einem Daten- oder Erneut-ausführen-Block gespeicherten Byte). Oracle unterstützt MD5-Prüfsummen. Weitere Informationen finden Sie in der Oracle-Dokumentation.

SQL Server-Instanzenname

Verfügbar, wenn als Datenbanktyp "SQLServer" ausgewählt wurde. Auf einem bestimmten Server, können Sie mehrere SQLServer-Services ausführen, jeder mit eigenen Ports, Anmeldungen und Datenbanken. Jeder dieser Services wird als eine Instanz von SQL Server bezeichnet. Dieses Feld gibt einen bestimmten Instanznamen für SQLServer an.

Hypersonic-Datenbanktyp

Verfügbar, wenn als Datenbanktyp "Hypersonic" ausgewählt wurde. Wählen Sie eine der folgenden Optionen in der Dropdown-Liste aus:

- Server
- Datei
- Speicherresident

Operator "Masseneinfügung in Datenbank"

Mit dem Operator "Masseneinfügung in Datenbank" können Sie schnell massenhaft Zeilen in eine Datenbanktabelle oder -ansicht basierend auf einem von Ihnen angegebenen Format importieren.

Eingabeparameter

Datenquelle

Der in der SQL-Anweisung anzugebende Tabellenname als Zeichenfolge oder Variable.

Zuordnungs-Array einfügen

Ein Array mit ValueMaps, die jeweils eine in die Datenbank einzufügende Zeile darstellen.

Wichtig! Stellen Sie sicher, dass Zeichenfolgenwerte durch einzelne Anführungszeichen eingekapselt sind.

Der ValueMap-Parametername, den Sie in eine Tabelle einfügen möchten, muss dem Namen des zugeordneten Spaltennamens entsprechen.

Alle Variablen innerhalb der Wertzuordnung müssen den Spalten in der Tabelle entsprechen. Wenn Sie beispielsweise über eine Tabelle mit den Spalten "Name" und "Nummer" verfügen, muss Ihre ValueMap auf dieselbe Weise organisiert sein.

Ausgabeparameter

Datenquelle

JDBCInsertMapArray

UserName Password

DatabaseType

OtherDatabaseType

DriverName

DatabaseName

CacheBufferSize

BatchPerfWorkaround

ConnectionProtocol

ServerType

DriverType

ServiceID

TNSName

ClientEncryption

ClientChecksum

InstanceName

HypersonicDatabaseType

ServerHost

ServerPort

ConnectionString

Operator "Aus Datenbank löschen"

Verwenden Sie den Operator "Aus Datenbank löschen", um auf der Grundlage von Kriterien, die Sie angeben, Zeilen in einer Tabelle zu löschen.

Eingabeparameter

Eingabequelle

Gibt an, dass der Anwender eine SQL-Anweisung als integrierten Ausdruck (Standardwert) oder Datenvariable senden kann. Wählen Sie aus der Dropdown-Liste entweder "Inline-Text" oder "Ausdruck" aus.

Inline-Text

Nur verfügbar, wenn "Inline-Text" als Eingabequelle ausgewählt ist. Gibt die generische SQL-Anweisung als Inline-Text an. Klicken Sie auf das Feld "Inline-Text", um den Inline-Text-Editor zu öffnen, wo Sie eine wörtliche SQL-Anweisung eingeben können.

Ausdruck

Nur verfügbar, wenn "Ausdruck" als Eingabequelle ausgewählt wird. Gibt die generische SQL-Anweisung als einen Ausdruck an. Verwenden Sie dieses Feld, um eine Variable anzugeben.

Eingabeparameter

Ein Array von Eingabewerten. Wenn die angegebene generische SQL-Anweisung die JDBC-Escape-Syntax verwendet und Eingabeparameter benötigt, können sie hier angegeben werden.

Ausgabeparameter

Abfrageergebnisse

Gibt die Anzahl der gelöschten Zeilen zurück

Ursache

Gibt den Grund an, wenn der Operator nach der Ausführung fehlschlägt.

Ergebnis

Gibt das Ergebnis der Operator-Ausführung an. InputSource

InlineText

InputSourceExpression

JDBCInputParamArray

IsConstructSQLStatement

CompleteSQLStatement

Datenquelle

SelectionCriteria

UserName

Kennwort

DatabaseType

OtherDatabaseType

DriverName

DatabaseName

CacheBufferSize

BatchPerfWorkaround

ConnectionProtocol

ServerType

DriverType

ServiceID

TNSName

ClientEncryption

ClientChecksum

InstanceName

HypersonicDatabaseType

ServerHost

ServerPort

ConnectionString

Operator "Datenbankschema abrufen"

Verwenden Sie den Operator "Datenbankschema abrufen", um Schemanamen von der Datenbank abzurufen.

Eingabeparameter

Die Anmeldeparameter für den Datenbankserver (siehe Seite 160) sind für diesen Operator erforderlich.

Ausgabeparameter

UserName

Kennwort

DatabaseType

OtherDatabaseType

DriverName

DatabaseName

CacheBufferSize

BatchPerfWorkaround

ConnectionProtocol

ServerType

DriverType

ServiceID

TNSName

ClientEncryption

ClientChecksum

InstanceName

HypersonicDatabaseType

ServerHost

ServerPort

ConnectionString

Operator "Freien Speicherplatz abrufen"

Verwenden Sie den Operator "Freien Speicherplatz abrufen", um den in der Datenbank verfügbaren freien Speicherplatz (in MB) zurückzugeben.

Eingabeparameter

Schemaname:

Gibt den Namen des Schemas an, für das freier Speicherplatz berechnet werden muss.

Ausgabeparameter

Abfrageergebnisse

Gibt den freien Speicherplatz (in MB) zurück.

Ursache

Gibt den Grund an, wenn der Operator nach der Ausführung fehlschlägt.

Ergebnis

Gibt das Ergebnis der Operator-Ausführung an.

SchemaName

UserName

Kennwort

DatabaseType

OtherDatabaseType

DriverName

DatabaseName

CacheBufferSize

BatchPerfWorkaround

ConnectionProtocol

ServerType

DriverType

ServiceID

TNSName

ClientEncryption

ClientChecksum

InstanceName

HypersonicDatabaseType

ServerHost

ServerPort

ConnectionString

Wenn die erforderlichen Eingabeparameter angegeben sind, wird ConnectionString automatisch erstellt.

Operator "Gespeicherte Prozedur abrufen"

Verwenden Sie den Operator "Gespeicherte Prozedur abrufen", um die gespeicherten verfügbaren Vorgangsnamen in der Datenbank zurückzugeben.

Eingabeparameter

Katalogname

Muss mit dem in der Datenbank gespeicherten Katalognamen übereinstimmen. Die Werte "" und Null zeigen an, dass der Katalogname nicht verwendet werden sollte, um die Suche einzuschränken.

Schemamuster

Muss mit dem in der Datenbank gespeicherten Schemanamen übereinstimmen. Die Werte "" und Null zeigen an, dass der Schemaname nicht verwendet werden sollte, um die Suche einzuschränken. Das Muster sollte ein regulärer Ausdruck sein.

Namensmuster für Verfahren

Muss mit dem in der Datenbank gespeicherten Verfahrensnamen übereinstimmen. Das Muster sollte ein von der Datenbank unterstützter regulärer Ausdruck sein.

Ausgabeparameter

Abfrageergebnisse

Gibt ein Array zurück, in dem jede Zeile ein Verfahrensname ist.

Ursache

Gibt den Grund an, wenn der Operator nach der Ausführung fehlschlägt.

Ergebnis

Gibt das Ergebnis des ausgeführten Operators an.

CatalogName

SchemaPattern

TableNamePattern

UserName

Kennwort

DatabaseType

OtherDatabaseType

DriverName

DatabaseName

CacheBufferSize

BatchPerfWorkaround

ConnectionProtocol

ServerTypeDriverType

ServiceID

TNSName

ClientEncryption

ClientChecksum

InstanceName

HypersonicDatabaseType

ServerHost

ServerPort

ConnectionString

Operator "Tabelle abrufen"

Verwenden Sie den Operator "Tabelle abrufen", um die Tabellenliste aus der Datenbank zurückzugeben.

Eingabeparameter

Katalogname

Der Katalogname muss mit dem in der Datenbank gespeicherten Katalognamen übereinstimmen. Die Werte "" und Null zeigen an, dass der Katalogname nicht verwendet werden sollte, um die Suche einzuschränken.

Schemamuster

Das Schemamuster muss mit dem in der Datenbank gespeicherten Schemanamen übereinstimmen. Die Werte "" und Null zeigen an, dass der Schemaname nicht verwendet werden sollte, um die Suche einzuschränken. Das Muster sollte ein regulärer Ausdruck sein.

Namensmuster für Tabelle

Das Tabellennamensmuster muss mit dem in der Datenbank gespeicherten Tabellennamen übereinstimmen. Das Muster sollte ein von der Datenbank unterstützter regulärer Ausdruck sein.

Ausgabeparameter

Abfrageergebnisse

Gibt ein Array zurück, in dem jede Zeile ein Tabellenname ist.

Ursache

Gibt den Grund an, wenn der Operator nach der Ausführung fehlschlägt.

Ergebnis

Gibt das Ergebnis des ausgeführten Operators an.

CatalogName

SchemaPattern

TableNamePattern

UserName

Kennwort

DatabaseType

OtherDatabaseType

DriverName

DatabaseName

CacheBufferSize

BatchPerfWorkaround

ConnectionProtocol

ServerType

DriverType

ServiceID

TNSName

ClientEncryption

ClientChecksum

InstanceName

HypersonicDatabaseType

ServerHost

ServerPort

ConnectionString

Operator "Verwendeten Speicherplatz abrufen"

Der Operator "Verwendeten Speicherplatz abrufen" gibt den in der Datenbank verwendeten Speicherplatz (in MB) zurück.

Eingabeparameter

Schemaname:

Gibt den Namen des Schemas an, für das die Operation den verwendeten Speicherplatz zurückgeben muss.

Ausgabeparameter

Abfrageergebnisse

Gibt den freien Speicherplatz (in MB) zurück.

Ursache

Gibt den Grund an, wenn der Operator nach der Ausführung fehlschlägt.

Ergebnis

Gibt das Ergebnis der Operator-Ausführung an.

SchemaName

UserName

Kennwort

DatabaseType

OtherDatabaseType

DriverName

DatabaseName

CacheBufferSize

BatchPerfWorkaround

ConnectionProtocol

ServerType

DriverType

ServiceID

TNSName

ClientEncryption

ClientChecksum

InstanceName

HypersonicDatabaseType

ServerHost

ServerPort

ConnectionString

Operator "Version abrufen"

Verwenden Sie den Operator "Version abrufen", um den Namen und die Versionsnummer der Datenbank zurückzugeben.

Eingabeparameter

Die Anmeldeparameter für den Datenbankserver (siehe Seite 160) sind für diesen Operator erforderlich.

Ausgabeparameter

UserName

Kennwort

DatabaseType

OtherDatabaseType

DriverName

DatabaseName

CacheBufferSize

BatchPerfWorkaround

ConnectionProtocol

ServerType

DriverType

ServiceID

TNSName

ClientEncryption

ClientChecksum

InstanceName

HypersonicDatabaseType

ServerHost

ServerPort

ConnectionString

Operator "Ansicht abrufen"

Verwenden Sie den Operator "Ansicht abrufen", um eine Liste von Ansichten aus der Datenbank zurückzugeben.

Eingabeparameter

Katalogname

Der Katalogname muss mit dem in der Datenbank gespeicherten Katalognamen übereinstimmen. Die Werte "" und Null zeigen an, dass der Katalogname nicht verwendet werden sollte, um die Suche einzuschränken.

Schemamuster

Das Schemamuster muss mit dem in der Datenbank gespeicherten Schemanamen übereinstimmen. Die Werte "" und Null zeigen an, dass der Schemaname nicht verwendet werden sollte, um die Suche einzuschränken. Das Muster sollte ein regulärer Ausdruck sein.

Namensmuster für Anzeige

Das Ansichtsnamensmuster muss mit dem in der Datenbank gespeicherten Ansichtsnamen übereinstimmen. Das Muster sollte ein regulärer Ausdruck sein.

Ausgabeparameter

Abfrageergebnisse

Gibt ein Array zurück, in dem jede Zeile ein Ansichtsname ist.

Ursache

Gibt den Grund an, wenn der Operator nach der Ausführung fehlschlägt.

Ergebnis

Gibt das Ergebnis des ausgeführten Operators an.

CatalogName

SchemaPattern

TableNamePattern

UserName

Kennwort

DatabaseType

OtherDatabaseType

DriverName

DatabaseName

CacheBufferSize

BatchPerfWorkaround

ConnectionProtocol

ServerType

DriverType

ServiceID

TNSName

ClientEncryption

ClientChecksum

InstanceName

HypersonicDatabaseType

ServerHost

ServerPort

ConnectionString

Operator "In Datenbank einfügen"

Verwenden Sie Operator "In Datenbank einfügen", um eine neue Zeile in einer Tabelle einzufügen.

Eingabeparameter

Eingabequelle

Gibt an, dass der Anwender eine SQL-Anweisung als integrierten Ausdruck (Standardwert) oder Datenvariable senden kann. Wählen Sie aus der Dropdown-Liste entweder "Inline-Text" oder "Ausdruck" aus.

Inline-Text

Nur verfügbar, wenn "Inline-Text" als Eingabequelle ausgewählt ist. Gibt die generische SQL-Anweisung als Inline-Text an. Klicken Sie auf das Feld "Inline-Text", um den Inline-Text-Editor zu öffnen, wo Sie eine wörtliche SQL-Anweisung eingeben können.

Ausdruck

Nur verfügbar, wenn "Ausdruck" als Eingabequelle ausgewählt wird. Gibt die generische SQL-Anweisung als einen Ausdruck an. Verwenden Sie dieses Feld, um eine Variable anzugeben.

Eingabeparameter

Ein Array von Eingabewerten. Wenn die angegebene generische SQL-Anweisung die JDBC-Escape-Syntax verwendet und Eingabeparameter benötigt, können sie hier angegeben werden.

Ausgabeparameter

Abfrageergebnisse

Gibt die Anzahl an Zeilen zurück, die in eine Tabelle eingefügt wurden.

Ursache

Gibt den Grund an, wenn der Operator nach der Ausführung fehlschlägt.

Ergebnis

Gibt das Ergebnis des ausgeführten Operators an.

InputSource

InlineText

InputSourceExpression

JDBCInputParamArray

IsConstructSQLStatement

CompleteSQLStatement

Datenquelle

ColumnNameMode

ColumnNames

ColumnNameAsArray

ColumnValueMode

ColumnValues

ColumnValueAsArray

UserName

Kennwort

DatabaseType

OtherDatabaseType

DriverName

DatabaseName

CacheBufferSize

BatchPerfWorkaround

ConnectionProtocol

ServerType

DriverType

ServiceID

TNSName

ClientEncryption

ClientChecksum

InstanceName

HypersonicDatabaseType

ServerHost

ServerPort

ConnectionString

Operator "Datenbank abfragen"

Verwenden Sie den Operator "Datenbank abfragen", um eine einzelne SQL-Anweisung für die Datenbank auszugeben. Dieser Operator unterstützt die JDBC-Escape-Syntax. Der Operator "Datenbank abfragen" verwendet "CallableStatement" und "ParameterMetaData", um Informationen über die Eingabe- und Ausgabeparameter vor und nach der Ausführung der Anweisung zu erfassen. Wenn der JDBC-Treiber dieses Verhalten nicht unterstützt, kann der generische SQL-Operator u. U. nicht alle Ergebnisse von den SQL-Anweisungen erfassen. Darüber hinaus unterstützt der vom Datenbankanbieter bereitgestellte JDBC-Treiber u. U. nicht die Verwendung aller Datentypen mit dem JDBC-Treiber. Zum Beispiel unterstützt der SQL-Server-JDBC-Treiber nicht den SQL-Server-Datentyp "sql_variant". Weitere Informationen zu den vom Treiber unterstützten JDBC-Funktionen und Datentypen finden Sie in der JDBC-Treiber-Dokumentation.

Eingabeparameter

Eingabequelle

Gibt an, dass der Anwender die SQL-Anweisung als integrierten Ausdruck (Standardwert) oder Datenvariable senden kann. Wählen Sie aus der Dropdown-Liste entweder "Inline-Text" oder "Ausdruck" aus.

Inline-Text

Nur verfügbar, wenn "Inline-Text" als Eingabequelle ausgewählt ist. Gibt die generische SQL-Anweisung als Inline-Text an.

Klicken Sie auf das Feld "Inline-Text", um den Inline-Text-Editor zu öffnen, wo Sie eine wörtliche SQL-Anweisung eingeben können.

Ausdruck

Nur verfügbar, wenn "Ausdruck" als Eingabequelle ausgewählt wird. Gibt die generische SQL-Anweisung als einen Ausdruck an. Verwenden Sie dieses Feld, um eine Variable anzugeben.

Höchstanzahl abzurufender Zeilen

Gibt die größtmögliche Anzahl von abzurufenden Zeilen an. Wenn leer, lautet der Standardwert 10 Zeilen. Der generische SQL-Operator ruft ein Maximum von 512 Zeilen ab. Weitere Zeilen werden abgeschnitten.

Eingabeparameter

Ein Array von Eingabewerten. Wenn die angegebene generische SQL-Anweisung die JDBC-Escape-Syntax verwendet und Eingabeparameter benötigt, können sie hier angegeben werden.

Nullwerte anzeigen

Wenn diese Option aktiviert ist, enthalten die Ergebnisse des generischen SQL-Operators das Feld "NullFieldFlags". Wenn die generische SQL-Anweisung Ergebnissätze zurückgibt, kann dieses Feld verwendet werden, um Nullwerte von Standardwerten zu unterscheiden.

Die Abfrage einer Ganzzahlspalte mit einem Nullwert gibt beispielsweise den Wert 0 zurück. Der Wert "NullFieldFlags" für diese Spalte wäre "Wahr". Wenn die Tabelle jedoch tatsächlich den Wert 0 speichern würde, würde der Wert "NullFieldFlags" "Falsch" lauten.

Ausgabeparameter

ResultsSets

Ein Array mit indizierten ValueMaps, in dem jedes Element die Ergebnisse einer Abfrage enthält. Die Größe von "ResultsSets" stimmt mit der Anzahl von Ergebnissätzen überein, die von der SQL-Anweisung zurückgegeben werden. Jede ValueMap enthält die folgenden Felder:

Zeilen

Ein Array mit ValueMaps, die die Zeilen des Ergebnissatzes repräsentieren. Jede ValueMap enthält ein Feld für jede Spalte und den Wert der Spalte in der betreffenden Zeile.

NullFieldFlags

Ein Array von "ValueMaps". Die Felder jeder ValueMap entsprechen den Feldern in Zeilen. Der Wert jedes Feldes ist "Wahr". Der entsprechende Wert in Zeilen lautet "Null" oder andernfalls "Falsch". Diese Ausgabe wird nur angezeigt, wenn "Nullwerte anzeigen" ausgewählt ist.

UpdatedRowCounters

Ein Array von Ganzzahlen, die die Anzahl von Zeilen repräsentieren, die mit der generischen SQL-Anweisung aktualisiert wurden. Wenn die generische SQL-Anweisung mehrere Aktualisierungen durchführt, enthält dieser Wert mehrere Werte.

RowCount

(Veraltet) Gibt entweder die Zeilenanzahl für SQL Data Manipulation Language (DML)-Anweisungen oder die unter "ResultSets" festgelegte Zeilenanzahl im ersten Ergebnissatz. Wenn die generische SQL-Anweisung keine Aktualisierungen durchführt und keine Ergebnissätze zurückgibt, dann wird dieser Wert auf -1 festgelegt. Dieses Feld wird für Rückwärtskompatibilität eingeschlossen.

OutputParam

Der Wert des Ausgabeparameters der generischen SQL-Anweisung. Wenn die generische SQL-Anweisung keine Ausgabeparameter zurückgibt, wird dieses Feld nicht in die Operator-Ergebnisse eingeschlossen. Die OutputParam-Felder werden nummeriert, zum Beispiel "outputParam1", "outputParam2".

isNullOutputParam

Ein boolescher Wert, der anzeigt, ob der entsprechende OutputParam-Wert Null ist. Dieses Feld wird nur angezeigt, wenn die generische SQL-Anweisung einen Ausgabeparameter zurückgibt und "Nullwerte anzeigen" ausgewählt ist.

Ursache

Gibt den Grund an, wenn der Operator nach der Ausführung fehlschlägt.

Ergebnis

Gibt das Ergebnis des ausgeführten Operators an.

InputSource

InlineText

InputSourceExpression

MaximumRows

JDBCInputParamArray

JDBCReportNull

UserName

Kennwort

DatabaseType

OtherDatabaseType

DriverName

DatabaseName

CacheBufferSize

BatchPerfWorkaround

ConnectionProtocol

ServerType

DriverType

ServiceID

TNSName

ClientEncryption

ClientChecksum

InstanceName

HypersonicDatabaseType

ServerHost

ServerPort

ConnectionString

Ausführen einer gespeicherten Prozedur

Um eine gespeicherte Prozedur für eine Datenbank auszuführen, verwenden Sie den Operator "Datenbank abfragen".

In diesem Beispiel wird der SQL Server-Treiber verwendet, der im Zuge der Installation von CA Process Automation angegeben wird. Um den Prozess für eine andere Datenbank auszuführen, laden Sie den entsprechenden JBDC-Treiber im Auswahlmenü "Anwenderressourcen verwalten" auf der Registerkarte "Konfiguration" hoch. CA Process Automation kann auf den Treiber zugreifen, wenn Sie den Koordinationsrechner-Service neu starten. Weitere Informationen finden Sie im *Handbuch für Inhaltsadministratoren*.

Sie können dieses Beispiel unter "SQL Server Management Studio" in der PAMReporting-Datenbank duplizieren. PAMReporting ist der Datenbankname, der im Zuge der Installation für die CA Process Automation-Berichterstellungstabellen angegeben wird. Sie können einen anderen Namen verwenden. Sie können jedoch auch eine beliebige Datenbank Ihrer Wahl verwenden.

Gehen Sie folgendermaßen vor:

1. Erstellen Sie eine gespeicherte Prozedur mit dem folgenden Text:

```
USE PAMReporting
GO
-- ============================================
-- Template generated from Template Explorer using:
-- Create Procedure (New Menu).SQL
--
-- Use the Specify Values for Template Parameters
-- command (Ctrl-Shift-M) to fill in the parameter
-- values below.
--
-- This block of comments will not be included in
-- the definition of the procedure.
-- ============================================
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE sp_getSOAPRows
       -- Add the parameters for the stored procedure here
AS
BEGIN
       -- SET NOCOUNT ON added to prevent extra result sets from
       -- interfering with SELECT statements.
       -- SET NOCOUNT ON;

    -- Insert statements for procedure here
          select count(*) from SOAPClientCall;
END
GO
```

2. Der Vorgang gibt die Anzahl von Zeilen in einer Tabelle zurück, die als SOAPClientCall aufgerufen wurde.
3. Klicken Sie in SQL Server Management Studio auf "Parse" (Analysieren).
4. Klicken Sie in SQL Server Management Studio auf "Execute" (Ausführen).

   CA Process Automation speichert die neue gespeicherte Prozedur in der PAMReporting-Datenbank.
5. Erstellen Sie einen Prozess mit dem Operator "Datenbank abfragen", und geben Sie anschließend die folgende Zeile in den Inline-Text des Operators ein:

   ```
   EXECUTE sp_getSOAPRows
   ```
6. Geben Sie in den Parametern der Datenbankserver-Anmeldung (siehe Seite 160) relevante Details ein, damit CA Process Automation mit Ihrer Datenbank kommunizieren kann.
7. Speichern Sie den Prozess.

8. Führen Sie den Prozess aus.
9. Öffnen Sie den Operatordatensatz.

   Die Anzahl der Zeilen, die die Prozedur zurückgibt, wird als Ergebnis der Ausführung angezeigt.

   Hinweis: Ihr Ergebnis hängt von der Anzahl der Zeilen in der Tabelle "SOAPClientCall" ab.

Operator "Aus Datenbank auswählen"

Verwenden Sie den Operator "Aus Datenbank auswählen", um mit optionalen Auswahlkriterien selektiv Daten aus einer oder mehreren Datenquellen abzurufen.

Eingabeparameter

Eingabequelle

Gibt an, dass der Anwender die SQL-Anweisung als integrierten Ausdruck (Standardwert) oder Datenvariable senden kann. Wählen Sie aus der Dropdown-Liste entweder "Inline-Text" oder "Ausdruck" aus.

Inline-Text

Nur verfügbar, wenn "Inline-Text" als Eingabequelle ausgewählt ist. Gibt die generische SQL-Anweisung als Inline-Text an. Klicken Sie auf das Feld "Inline-Text", um den Inline-Text-Editor zu öffnen, wo Sie eine wörtliche SQL-Anweisung eingeben können.

Ausdruck

Nur verfügbar, wenn "Ausdruck" als Eingabequelle ausgewählt wird. Gibt die generische SQL-Anweisung als einen Ausdruck an. Verwenden Sie dieses Feld, um eine Variable anzugeben.

Höchstanzahl abzurufender Zeilen

Gibt die größtmögliche Anzahl an Zeilen an, die mittels der Auswahl-Anweisung abgerufen werden sollen. Dieser Parameter überschreibt die auf Operatorkategorienebene festgelegte Eigenschaft.

Eingabeparameter

Ein Array von Eingabewerten. Wenn die angegebene generische SQL-Anweisung die JDBC-Escape-Syntax verwendet und Eingabeparameter benötigt, können sie hier angegeben werden.

Nullwerte anzeigen

Wenn diese Option aktiviert ist, enthalten die Ergebnisse des generischen SQL-Operators das Feld "NullFieldFlags". Wenn die generische SQL-Anweisung Ergebnissätze zurückgibt, kann dieses Feld verwendet werden, um Nullwerte von Standardwerten zu unterscheiden.

Die Abfrage einer Ganzzahlspalte mit einem Nullwert gibt beispielsweise den Wert 0 zurück. Der Wert "NullFieldFlags" für diese Spalte ist "Wahr". Wenn in der Tabelle jedoch tatsächlich der Wert 0 gespeichert wäre, würde der Wert "NullFieldFlags" "Falsch" lauten.

Ausgabeparameter

Ergebnis der Abfrage

Ein Array mit ValueMaps, die die Zeilen des Ergebnissatzes repräsentieren. Jede ValueMap enthält ein Feld für jede Spalte und den Wert der Spalte in der betreffenden Zeile.

NullFieldFlags

Ein Array von "ValueMaps". Die Felder jeder ValueMap entsprechen den Feldern in Zeilen. Der Wert der einzelnen Felder ist entweder "Wahr" (der zugehörige Wert in den Zeilen ist Null) oder "Falsch" (das zugehörige Feld ist nicht Null). Diese Ausgabe wird nur angezeigt, wenn "Nullwerte anzeigen" ausgewählt ist.

InputSource

InlineText

InputSourceExpression

MaximumRows

JDBCInputParamArray

JDBCReportNull

IsConstructSQLStatement

CompleteSQLStatement

ReturnValueMode ReturnValues

ReturnValuesAsArray

DataSourceMode

DataSources

DataSourcesAsArray

SelectionCriteria

SortCriteriaMode

SortCriteria

SortCriteriaAsArray

UserName Password

DatabaseType

OtherDatabaseType

DriverName

DatabaseName

CacheBufferSize

BatchPerfWorkaround

ConnectionProtocol

ServerType

DriverType

ServiceID

TNSName

ClientEncryption

ClientChecksum

InstanceName

HypersonicDatabaseType

ServerHost

ServerPort

ConnectionString

Operator "In Datenbank aktualisieren"

Verwenden Sie den Operator "In Datenbank aktualisieren", um Datensätze in einer Tabelle zu aktualisieren.

Eingabeparameter

Eingabequelle

Gibt an, dass der Anwender eine SQL-Anweisung als integrierten Ausdruck (Standardwert) oder Datenvariable senden kann. Wählen Sie aus der Dropdown-Liste entweder "Inline-Text" oder "Ausdruck" aus.

Inline-Text

Nur verfügbar, wenn "Inline-Text" als Eingabequelle ausgewählt ist. Gibt die generische SQL-Anweisung als Inline-Text an. Klicken Sie auf das Feld "Inline-Text", um den Inline-Text-Editor zu öffnen, wo Sie eine wörtliche SQL-Anweisung eingeben können.

Ausdruck

Nur verfügbar, wenn "Ausdruck" als Eingabequelle ausgewählt wird. Gibt die generische SQL-Anweisung als einen Ausdruck an. Verwenden Sie dieses Feld, um eine Variable anzugeben.

Eingabeparameter

Ein Array von Eingabewerten. Wenn die angegebene generische SQL-Anweisung die JDBC-Escape-Syntax verwendet und Eingabeparameter benötigt, können sie hier angegeben werden.

Ausgabeparameter

Abfrageergebnisse

Gibt die Anzahl der aktualisierten Zeilen zurück.

Ursache

Gibt den Grund an, wenn der Operator nach der Ausführung fehlschlägt.

Ergebnis

Gibt die Ergebnisse des ausgeführten Operators an.

InputSource

InlineText

InputSourceExpression

JDBCInputParamArray

IsConstructSQLStatement

CompleteSQLStatement

FieldValueModeArray

FieldValueModeC2OValueMap

Datenquelle

FieldsValues

FieldsAsArray

ValuesAsArray

FieldValueMap

SelectionCriteria

UserName

Kennwort

DatabaseType

OtherDatabaseType

DriverName

DatabaseName

CacheBufferSize

BatchPerfWorkaround

ConnectionProtocol

ServerType

DriverType

ServiceID

TNSName

ClientEncryption

ClientChecksum

InstanceName

HypersonicDatabaseType

ServerHost

ServerPort

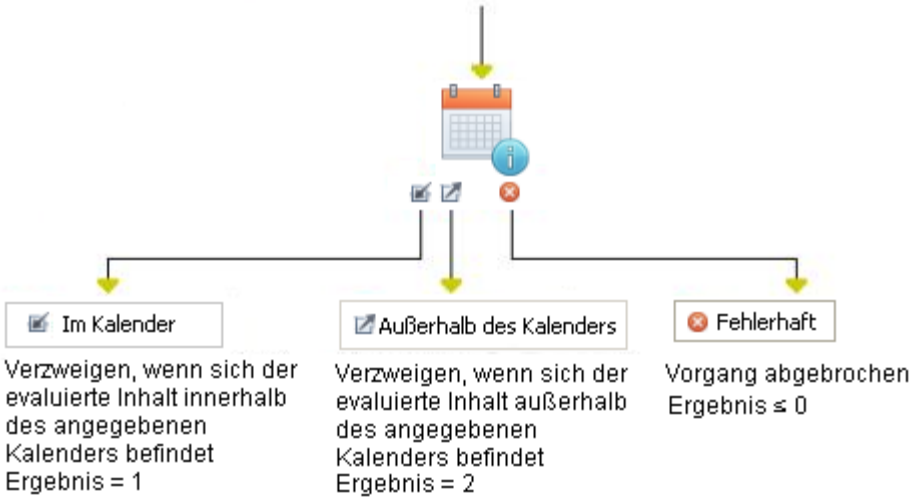
ConnectionString

Kapitel 6: Datum - Uhrzeit

Die Operatoren "Datum - Uhrzeit" sind für die Einhaltung von Uhrzeit- und Kalendereinschränkungen in Prozessen zuständig Alle Operatoren dieser Gruppe können nur auf einem Koordinationsrechner ausgeführt werden.

Operator "Kalender überprüfen"

Der Operator "Kalender überprüfen" gibt an, ob ein Datum in einen Satz von Kalenderregeln fällt.


Im Kalender
Verzweigen, wenn sich der evaluierte Inhalt innerhalb des angegebenen Kalenders befindet
Ergebnis = 1
Außerhalb des Kalenders
Verzweigen, wenn sich der evaluierte Inhalt außerhalb des angegebenen Kalenders befindet
Ergebnis = 2
Fehlerhaft
Vorgang abgebrochen
Ergebnis ≤ 0

Eingabeparameter

Kalender mit erlaubten Daten

Gibt den vollständigen Pfad des Kalenders an. Dieser Ausdruck definiert die zulässigen Daten für nachfolgende Verzweigungen in dem zu verarbeitenden Prozess.

Klicken Sie, um ein Kalenderobjekt zu finden. Nachdem Sie einen Kalender ausgewählt haben, klicken Sie auf "Öffnen", um das Kalenderobjekt im Kalenderdesigner zu öffnen.

Kalender mit ausgeschlossenen Daten

Gibt den vollständigen Pfad des Kalenders an, der zur Definition ausgeschlossener Tage verwendet wird. Ausgeschlossene Tage sind Tage, wenn nachfolgende Verzweigungen im Prozess nicht verarbeitet werden dürfen.

Es gibt keine ausgeschlossenen Tage, wenn kein Kalender angegeben wird.

Klicken Sie, um ein Kalenderobjekt zu finden. Nachdem Sie einen Kalender ausgewählt haben, klicken Sie auf "Öffnen", um das Kalenderobjekt im Kalenderdesigner zu öffnen.

Delta

Gibt die Anzahl von Tagen an, um die ein zulässiges Datum verschoben wird, wenn es auf ein ausgeschlossenes Datum fällt.

Die Verschiebung hängt davon ab, ob der Wert positiv, negativ oder Null ist. Ein negativer Wert bewirkt eine Verschiebung nach vorn (früher) und ein positiver Wert eine Verschiebung nach hinten (später). Wenn dieser Wert Null ist, wird das zulässige Datum übersprungen.

Nur offene Tage

Zählt beim Verschieben eines gültigen Datums, das auf ein ausgeschlossenes Datum fällt, nur offene Tage.

Offene Tage sind Tage, die keiner Bedingung oder Regel entsprechen, mit der Daten geschlossen oder ausgeschlossen werden.

Maximale Aufschübe

Diese Option definiert die größtmögliche Anzahl an Aufschüben, die zulässig sind, wenn nachfolgende Aufschübe auf geschlossene Daten fallen.

Diese Einstellung ist nur relevant, wenn "Nur offene Tage" aktiviert ist.

Datum

Gibt das Datum an, das mit den Kalenderregeln getestet wird.

Klicken Sie auf das Kalendersymbol, um ein Datum auszuwählen. Diese Option ist nicht verfügbar, wenn Sie das Kontrollkästchen "Aktuelles Datum verwenden" oder "Berechnetes Datum verwenden" wählen.

Aktuelles Datum verwenden

Gibt an, das aktuelle Datum zu verwenden, um es mit den Kalenderregeln zu testen.

Deaktivieren Sie dieses Kontrollkästchen, um ein bestimmtes Datum im Feld "Datum" anzugeben. Dieses Kontrollkästchen ist standardmäßig ausgewählt.

Berechnetes Datum verwenden

Gibt an, das Datum aus dem Feld "Berechnetes Datum" zu verwenden.

Berechnetes Datum

Wenn "Berechnetes Datum verwenden" ausgewählt ist, gibt dieser Parameter ein Datum zurück. Ein typischer Einsatzbereich ist die Berechnung eines zukünftigen Datums basierend auf dem aktuellen Datum.

Ausgabeparameter

CalenderDate

VacationsDate

Delta

OpenDays

MaxShifts

Datum

UseCurrentDate

UseCalculatedDate

CalculatedDate

Operator "Datum/Uhrzeit überprüfen"

Der Operator "Datum- und Uhrzeitprüfung" führt Verzweigungen in einem Prozess bedingt danach aus, ob ein angegebenes Datum und die angegebene Uhrzeit überschritten wurden. Der Operator "Datum- und Uhrzeitprüfung" platziert die folgende Uhrzeitbedingung in einem Prozess.

Eine Uhrzeitbedingung erlaubt die Bearbeitung, um mit nachfolgenden Verzweigungen in einem Prozess vor und nach dem in den Operator-Eigenschaften festgelegten Datum fortzufahren. Der Operator kann verwendet werden, um Datums- und Uhrzeitbedingungen auf unterschiedliche Segmente von Prozessen zu platzieren, die mehrmals am Tag ausgeführt werden. Mit dieser Einstellung können Sie zusätzliche Verknüpfungen auf dem Operator hinzufügen, um Verzweigungen anzugeben, die vor (<) oder nach (>) einem Datum und einer Zeit bearbeitet werden sollen.

Im Gegensatz dazu wird das Kontrollkästchen "Angegebenes Datum und Uhrzeit abwarten" anfänglich für die Bedingung "Datum/Uhrzeit abwarten" ausgewählt. Der Operator bearbeitet dann nur zusätzliche "nach (>)"-Verknüpfungen, während zusätzliche "vor (<)"-Verknüpfungen ignoriert werden.

Eingabeparameter

Datum

Gibt das Datum an, an dem die Zulässigkeit für die Bearbeitung von nachfolgenden Verzweigungen im Prozess bestimmt werden soll.

Klicken Sie auf das Symbol "Kalender", um den Kalender zu öffnen und ein Datum auszuwählen. Diese Option ist nicht verfügbar, wenn Sie das Kontrollkästchen "Aktuelles Datum verwenden" oder "Berechnetes Datum verwenden" wählen.

Aktuelles Datum verwenden

Gibt das Datum an, an dem der Prozess ausgeführt werden soll, um die Zulässigkeit für das Bearbeiten von nachfolgenden Verzweigungen im Prozess zu bestimmen.

Deaktivieren Sie dieses Kontrollkästchen, um ein bestimmtes Datum im Feld "Datum" anzugeben. Dieses Kontrollkästchen ist standardmäßig ausgewählt.

Berechnetes Datum verwenden

Gibt ein Datum zurück. In diesem Ausdruck können Sie die Datumsvariablen und -funktionen von CA Process Automation verwenden, um ein Datum zurückzugeben.

Berechnetes Datum

Wenn "Berechnetes Datum verwenden" ausgewählt ist, gibt dieser Parameter einen Ausdruck an, der ein Datum zurückgibt. Ein typischer Einsatzbereich ist die Berechnung eines zukünftigen Datums basierend auf dem aktuellen Datum.

Uhrzeit

Gibt eine Zeit in einem 12-Stunden-Format (HH:MM PM/AM) an.

Beispiel: 07:30:00 PM

Angegebenes Datum und Uhrzeit abwarten

Erstellt eine "Datum/Uhrzeit abwarten"-Bedingung. Diese Eigenschaft verzögert die Verarbeitung von nachfolgenden Operatoren in einer Verzweigung des Prozesses bis zur angegebenen Uhrzeit. Es werden nur Ausgangsverknüpfungen verarbeitet, die planmäßig nach (>) der angegebenen Zeit auftreten.

Deaktivieren Sie dieses Kontrollkästchen, um die Verarbeitung auf unterschiedliche Verzweigungen vor oder nach dem angegebenen Datum und der festgelegten Zeit umzulenken. Der Operator wendet die folgenden Bedingungen an, wenn das Kontrollkästchen "Angegebenes Datum und Uhrzeit abwarten" deaktiviert wird:

Wenn das angegebene Datum und die Uhrzeit in der Zukunft liegen, werden nur Ausgangsverknüpfungen bearbeitet, die *vorher* (<) bearbeitet werden sollen.

Ausgabeparameter

Datum

UseCurrentDate

UseCalculatedDate

CalculatedDate

Uhrzeit

WaitForSpecifiedDate

Kapitel 7: Verzeichnisdienste

Die Verzeichnisdienste-Operatoren bieten eine Schnittstelle, um *Lightweight Directory Access Protocol* (LDAP) zu unterstützen. Sie können die Vorgänge automatisieren, die unter LDAP-Servern ausgeführt werden. Mit Ausnahme der Active Directory-spezifischen Operatoren arbeiten alle Operatoren mit verschiedenen LDAP-Servern. Ungeachtet der Betriebssystemplattform, auf der CA Process Automation ausgeführt wird, sollten die Verzeichnisdienste-Operatoren auf einem CA Process Automation Automation-Koordinationsrechner oder Agenten mit den gleichen Ergebnissen ausgeführt werden.

LDAP-Anmeldeparameter

Die auf Verzeichnisdienste-Kategorienebene angegebenen Standard-LDAP-Felder können auf der LDAP-Anmeldeparameter-Seite überschrieben werden. Diese Seite ist Teil der Eingabe für jeden Operator "Verzeichnisdienste". Wenn ein Feld einen Wert enthält, wird damit der Wert überschrieben, der für das gleichnamige Feld auf der Kategoriekonfigurationsebene angegebenen wurde.

Remote-LDAP-Host

Gibt die URL oder IP-Adresse des LDAP-Servers an.

Remote-LDAP-Serverport

Gibt den LDAP-Serverport an.

LDAP-Anwender

Gibt an, dass sich der LDAP-Anwender, der Zugriff auf den LDAP-Server hat, anmelden können sollte. Allerdings werden die Operationen, die von diesem Anwender ausgeführt werden können, durch die für die LDAP-Einträge festgelegten ACIs beschränkt.

LDAP-Kennwort für Anwender

Gibt das Kennwort für den LDAP-Anwender an.

Basis-DN

Gibt den zu verwendenden standardmäßigen Basis-DN (Distinguished Name) an. Dies ist die Basis-DN, unter dem sich der LDAP-Anwender befindet.

Anwenderpräfix

Gibt das zu verwendende Anwenderpräfix an, das entweder "uid" oder "cn" sein kann.

Operator "Computer zu Domäne hinzufügen"

Der Operator "Computer zu Domäne hinzufügen" dient zum Erstellen eines neuen Computerobjekts im Active Directory-Server. Dieser Operator bezieht sich nur auf einen Active Directory-Server.

Eingabeparameter

Computerpfad

Gibt den Distinguished Name des Objekts an, unter dem Sie das neue Computerobjekt erstellen wollen.

Computername

Gibt den Namen des neuen Computerobjekts an.

Ausgabeparameter

LDAPADComputerBaseDn

LDAPADComputerName

remoteLDAPHost

remoteLDAPPort

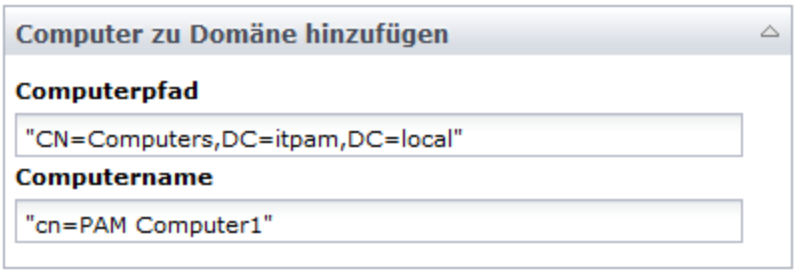
remoteLDAPUser

remoteLDAPPassword

LDAPBaseDN

LDAPUserPrefix

Beispiel



Fügen Sie das LDAP-Attribut zum Namen des neuen Computerobjekts hinzu. Computerobjekte verwenden normalerweise das Attribut "cn" als Teil des RDN (relativer Distinguished Name) des Computernamens.

Operator-Fehler

Dieser Operator wird in den folgenden Fällen fehlschlagen:

- Der Name des neuen Computerobjekts wird bereits verwendet.
- Einige der obligatorischen Attribute, die zum Erstellen des neuen Computerobjekts erforderlich sind, fehlen.
- Der Pfad, unter dem das Computerobjekt erstellt werden soll, ist ungültig.
- Unfähig, eine Verbindung mit dem Active Directory-Server herzustellen.

Operator "Anwender zur Gruppe hinzufügen"

Der Operator "Anwender zur Gruppe hinzufügen" fügt einer LDAP-Gruppe auf einem LDAP-Server einen LDAP-Anwender hinzu.

Eingabeparameter

Anwender-DN

Gibt den Distinguished Name des Anwenders an, der der Gruppe hinzugefügt werden soll.

Group DN

Gibt den Distinguished Name der Gruppe an, der der Anwender hinzugefügt werden soll.

Ausgabeparameter

LDAPUserDn

LDAPGroupDn

remoteLDAPHost

remoteLDAPPort

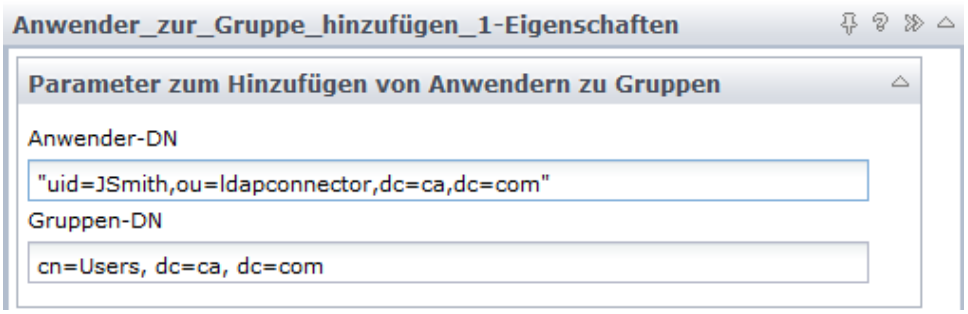
remoteLDAPUser

remoteLDAPPassword

LDAPBaseDN

LDAPUserPrefix

Beispiel



Operator-Fehler

Dieser Operator wird in den folgenden Fällen fehlschlagen:

- Der Anwender ist bereits ein Mitglied der Gruppe.
- Die Gruppe existiert nicht im LDAP-Server.
- Der DN der Gruppe zeigt auf ein LDAP-Objekt, das nicht vom Typ group, groupofnames oder groupofuniquenames ist.
- Unfähig, eine Verbindung mit dem LDAP-Server herzustellen.

Operator "Gruppe erstellen"

Der Operator "Gruppe erstellen" erstellt ein LDAP-Gruppenobjekt auf dem LDAP-Server. Dieses Gruppenobjekt kann vom folgenden Typ sein: "Gruppe", "Namensgruppe" oder "Gruppe eindeutiger Namen".

Eingabeparameter

Gruppenpfad

Gibt den Distinguished Name des Objekts an, unter dem Sie das neue Gruppenobjekt erstellen wollen.

Gruppenname

Gibt den Namen der Gruppe an, die Sie erstellen wollen. Dies ist das CN-Attribut der Gruppe.

LDAP-Gruppentyp

Gibt den Typ der LDAP-Gruppe an. Wählen Sie "Gruppe", "Namensgruppe" oder "Gruppe eindeutiger Namen" aus.

Angegebenes Array-Feld für die Gruppenmitglieder verwenden

Wenn dies aktiviert ist, wird für diese Anfrage das Mitglieder-Array verwendet.

LDAP-Gruppenmitglieder-Array

Gibt das Array mit Mitgliedern der Gruppe an (erforderlich für "Namensgruppe" oder "Gruppe eindeutiger Namen" in Active Directory). Dieses Feld wird nur aktiviert, wenn das Feld "Angegebenes Array-Feld für die Gruppenmitglieder verwenden" aktiviert ist.

LDAP-Gruppenmitglieder

Gibt die Mitglieder der Gruppe an (erforderlich für "Namensgruppe" oder "Gruppe eindeutiger Namen in Active Directory). Dieses Feld wird nur aktiviert, wenn das Feld "Angegebenes Array-Feld für die Gruppenmitglieder verwenden" deaktiviert ist.

Wird ein Objekt des Typs 'Gruppe' in Active Directory erstellt?

Muss aktiviert werden, wenn wir ein Objekt vom Typ "Gruppe" erstellen, wie unter "LDAP-Gruppentyp" in einem Active Directory-Server angegeben wurde. Für ein Objekt "Gruppe" in Active Directory können Sie zwei zusätzliche Attribute festlegen: "Umfang der Active Directory-Gruppe" und "Typ der Active Directory-Gruppe".

Umfang der Active Directory-Gruppe

Gibt den Umfang der in Active Directory erstellten Gruppe an. Wählen Sie entweder "Domäne (lokal)", "Global" oder "Universal" aus. Dieses Feld wird nur aktiviert, wenn Sie das Kontrollkästchen "Wird ein Objekt des Typs 'Gruppe' in Active Directory erstellt?" aktivieren.

Typ der Active Directory-Gruppe

Gibt den Typ der in Active Directory erstellten Gruppe an. Wählen Sie entweder "Sicherheit" oder "Verteilung" aus. Dieses Feld wird nur aktiviert, wenn das Feld "Wird ein Objekt des Typs 'Gruppe' in Active Directory erstellt?" aktiviert ist.

Hinweise

- Im Active Directory kann keine Gruppe mit dem Umfang "Universal" vom Typ "Sicherheit" sein. Der Betriebsablauf schlägt in diesem Fall fehl.
- Die Werte, die in den Feldern "Umfang der Active Directory-Gruppe" und "Typ der Active Directory-Gruppe" angegeben sind, werden ignoriert, wenn das Feld "LDAP-Gruppentyp" nicht auf den Wert "Gruppe" festgelegt ist.
- Die Felder "Umfang der Active Directory-Gruppe" und "Typ der Active Directory-Gruppe" werden nicht ignoriert, wenn "LDAP-Gruppentyp" nicht auf den Wert "Gruppe" gesetzt wurde. Diese Felder werden aktiviert, wenn Sie das Kontrollkästchen "Wird ein Objekt des Typs "Gruppe" in Active Directory erstellt?" aktivieren (da in CA Process Automation keine Mehrfachauswahl ausgeführt werden kann). Die Werte dieser Felder sind nur relevant, wenn Sie in Active Directory eine "Gruppe" erstellen. In der Anwenderoberfläche sind sie u. U.noch aktiv, aber im Backend werden sie in allen anderen Fällen ignoriert.

Ausgabeparameter

LDAPGroupBaseDn

LDAPGroupName

LDAPGroupType

LDAPGroupMembersType

LDAPGroupMembersArray

LDAPGroupMembers

LDAPIsADGroup

LDAPADGroupScope

LDAPADGroupType

remoteLDAPHost

remoteLDAPPort

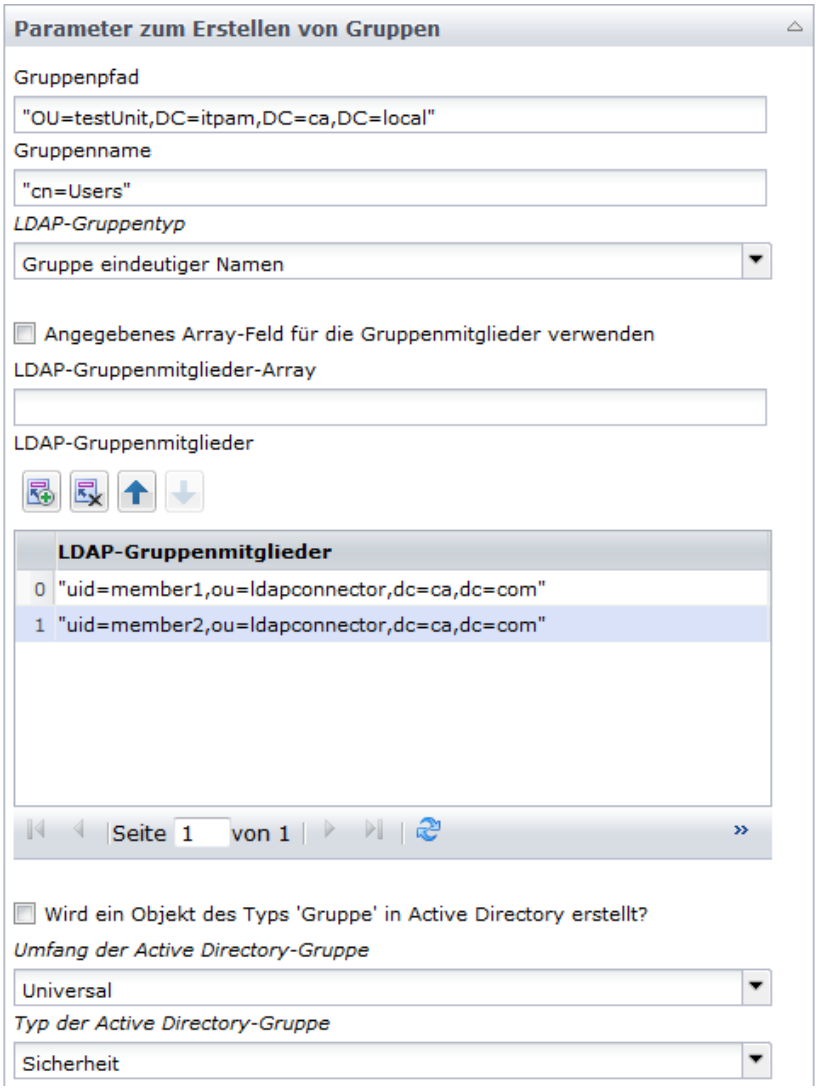
remoteLDAPUser

remoteLDAPPassword

LDAPBaseDN

LDAPUserPrefix

Beispiel



Operator-Fehler

Dieser Operator wird in den folgenden Fällen fehlschlagen:

- Die Gruppe existiert schon.
- Einige der obligatorischen Attribute, die zum Erstellen des neuen Gruppenobjekts erforderlich sind, fehlen.
- Der Pfad, unter dem das neue Gruppenobjekt erstellt werden soll, ist ungültig.
- Der LDAP-Server unterstützt nicht den angegebenen Gruppentyp: "Gruppe", "Namensgruppe" oder "Gruppe eindeutiger Namen".

- Die Gruppe wird in einem Active Directory-Server mit dem Umfang "Universal" und dem Typ "Sicherheit" erstellt.
- Das Objekt wird in Active Directory als eine "Namensgruppe" erstellt, ohne dass im Betriebsablauf Mitglieder festgelegt wurden.
- Das Objekt wird in Active Directory als eine "Gruppe eindeutiger Namen" erstellt, ohne dass im Betriebsablauf Mitglieder festgelegt wurden.
- Unfähig, eine Verbindung mit dem LDAP-Server herzustellen.

Operator "Objekt erstellen"

Der Operator "Objekt erstellen" erstellt ein LDAP-Objekt eines beliebigen Typs auf dem LDAP-Server.

Eingabeparameter

Objektpfad

Gibt den Distinguished Name des Objekts an, unter dem Sie das neue LDAP-Objekt erstellen wollen.

Objektname

Gibt den Namen des neuen LDAP-Objekts an.

Stellen Sie sicher, dass Sie das LDAP-Attribut dem Namen des neuen LDAP-Objekts hinzufügen. Das Attribut könnte "ou", "cn", "uid" usw. sein und hängt vom Typ des LDAP-Objekts ab, das erstellt wird.

Angegebenes Array-Feld für die "objectclass"-Attributwerte des Objekts verwenden

Wenn aktiviert, wird für diese Anfrage das Attributwerte-Array "objectclass" verwendet.

Array mit "objectclass"-Attributwerten des Objekts

Gibt das die Werte des "objectclass"-Attributs enthaltende Array an. Dieses Datensatzfeld muss als Array (indizierte Zeichenfolge) definiert werden. Wenn "Angegebenes Array-Feld für die "objectclass"-Attributwerte des Objekts verwenden" aktiviert ist, wird dieses Feld verwendet.

"objectclass"-Attributwerte des Objekts

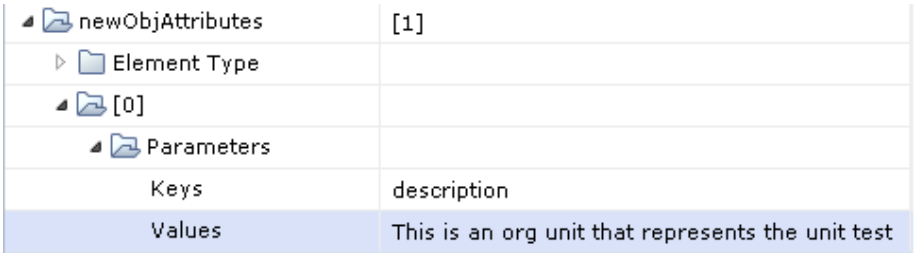
Gibt die Werte des "objectclass"-Attributs an. Wenn "Angegebenes Array-Feld für die "objectclass"-Attributwerte des Objekts verwenden" deaktiviert ist, wird dieses Feld verwendet.

"objectclass" ist das LDAP-Attribut, das den Typ des neuen Objekts definiert.

Wertzuordnungen mit zusätzlichen Objekt-LDAP-Attributen

Dies ist ein Array von Wertzuordnungen, das zusätzliche für das neue Objekt festzulegende LDAP-Attribute enthält. Der Schlüssel jeder Wertzuordnung muss vom Typ "Zeichenfolge" sein; der Wert muss vom Typ "Zeichenfolge" oder "Zeichenfolgen-Array" (indizierte Zeichenfolge) sein. Der Schlüssel muss "Key" genannt werden, und der Wert muss "Values" genannt werden.

Der Anwender kann das Feld "Values" auf den Typ "Zeichenfolge" festlegen, um einwertige LDAP-Attribute für das neu erstellte LDAP-Objekt zu erstellen. Zum Beispiel:



Das Objekt "newObjAttributes" eine indizierte Wertzuordnung, deren Schlüsselfelder "Keys" genannt werden und vom Typ "Zeichenfolge" sind und deren Wertfelder "Values" genannt werden und vom Typ "Zeichenfolge" sind.

Alternativ kann der Anwender das Feld "Values" als Typ "Zeichenfolgen-Array" (indizierte Zeichenfolge) festlegen, um mehrwertige LDAP-Attribute für das neu erstellte LDAP-Objekt zu erstellen.

Das Objekt "newObjAttributes2" eine indizierte Wertzuordnung, deren Schlüsselfelder "Keys" genannt werden und vom Typ "Zeichenfolge" sind und deren Wertfelder "Values" genannt werden und indizierte Zeichenfolgen sind. In diesem Fall kann der Anwender sowohl einwertige als auch mehrwertige LDAP-Attribute für das neue erstellte LDAP-Objekt erstellen.

Zum Beispiel:

newObjAttributes2	[2]
Element Type	
[0]	
Parameters	
Keys	telephonenumber
Values	[2]
[0]	555-55-5555
[1]	555-55-0000
[1]	
Parameters	
Keys	description
Values	[1]
[0]	The test unit organization

Innerhalb des gleichen Objekts "newObjAttributes2" ein mehrwertiges Attribut "telephonenumber" und ein einwertiges Attribut "description".

Beachten Sie, dass nur der letzte mit dem Schlüssel verbundene Wert erhalten bleibt, wenn der gleiche Schlüssel mehrmals innerhalb der indizierten ValueMap angezeigt wird.

Beachten Sie, dass die unter "Wertzuordnungen mit zusätzlichen Objekt-LDAP-Attributen" eingegebenen Attributnamen die LDAP-Namen dieser Attribute sein müssen, wie im LDAP-Server-Schema angegeben. Um zum Beispiel den Wert des Attributs "Nachname" festzulegen, müssen Sie den LDAP-Namen dieses Attributs verwenden: "sn", um den Wert des Attributs "Vorname" festzulegen, Sie müssen das Attribut "givenname" usw. verwenden. Siehe Allgemeine LDAP-Attributnamen (siehe Seite 237).

Die LDAP-Namen unterscheiden sich von den Attributanzeigenamen.

Die meisten LDAP-Server unterscheiden sich in den Anzeigenamen der LDAP-Attribute, aber sie alle müssen die LDAP-Namen dieser Attribute unterstützen. Aus diesem Grund müssen die LDAP-Namen der Attribute statt der Anzeigenamen verwendet werden.

Ausgabeparameter

LDAPCreateObjectBaseDn

LDAPCreateObjectName

LDAPCreateObjObjectClassUseArray

LDAPCreateObjObjectClassArray

LDAPCreateObjObjectClass

LDAPCreateObjectAttributes

remoteLDAPHost

remoteLDAPPort

remoteLDAPUser

remoteLDAPPassword

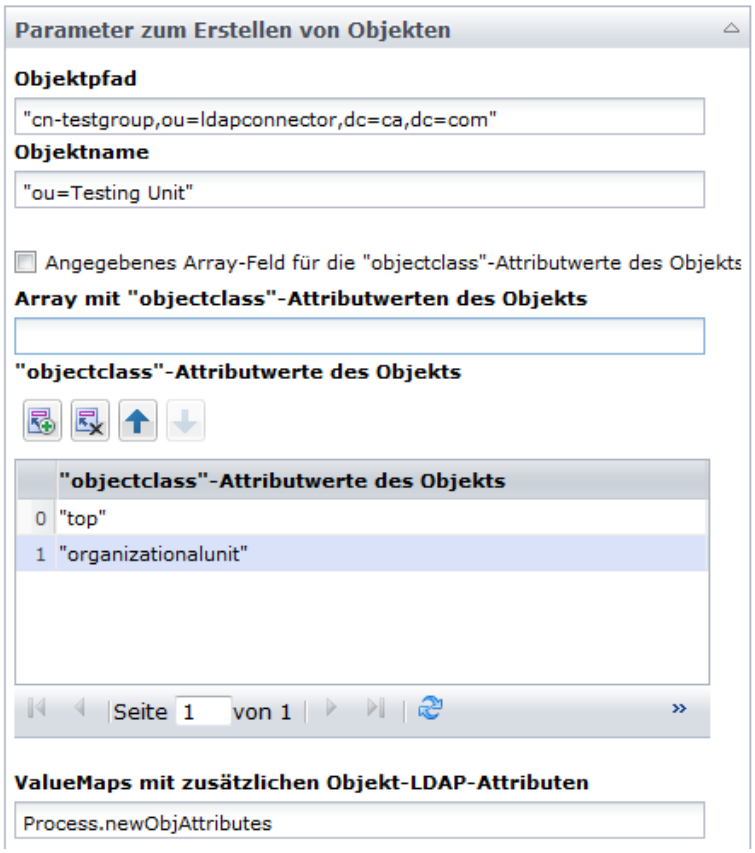
LDAPBaseDN

LDAPUserPrefix

Beispiel

In diesem Beispiel erstellen wir eine Organisationseinheit mit dem Namen "Testeinheit". Das Attribut "objectclass" gibt an, dass das Objekt vom Typ "Oben" und "Organisationseinheit" ist. Der Typ "Oben" ist der Stamm aller LDAP-Typen.

Wir fügen der neuen Organisationseinheit auch zusätzliche Attribute mittels der indizierten Wertzuordnung "Process.newObjAttributes" hinzu.



Operator-Fehler

Dieser Operator wird in den folgenden Fällen fehlschlagen:

- Der Name des neuen LDAP-Objekts wird schon verwendet.
- Einige der obligatorischen Attribute, die zum Erstellen eines neuen LDAP-Objekts benötigt werden, fehlen.
- "objectclass" des neuen LDAP-Objekts fehlt oder ist falsch.
- Einige Attribute, die für das Objekt erstellt werden, enthalten ungültige Werte.
- Einige Attribute, die für das Objekt erstellt werden, gelten nicht für diesen Objekttyp, zum Beispiel können Sie einem LDAP-Objekt des Typs "Organisationseinheit" kein "Mail"-Attribut hinzufügen.
- Der Pfad, unter dem das LDAP-Objekt erstellt werden soll, ist ungültig.

- Der Anwender stellt sicher, dass für das Attribut "objectclass" ein Attribute-Array verwendet wird, aber das ins Array-Feld eingegebene CA Process Automation-Objekt ist eigentlich nicht vom Typ "Array" (indizierte Zeichenfolgen).
- Unfähig, eine Verbindung mit dem LDAP-Server herzustellen.

Operator "Organisationseinheit erstellen"

Der Operator "Organisationseinheit erstellen" erlaubt einem Anwender, ein LDAP-Objekt vom Typ "Organisationseinheit" auf dem LDAP-Server zu erstellen.

Eingabeparameter

Pfad der Organisationseinheit

Gibt den Distinguished Name des Objekts an, unter dem das neue Organisationseinheit-Objekt erstellt werden soll.

Name der Organisationseinheit

Gibt den Namen des neuen Organisationseinheit-Objekts an.

Ausgabeparameter

LDAPOrgUnitBaseDn

LDAPOrgUnitName

remoteLDAPHost

remoteLDAPPort

remoteLDAPUser

remoteLDAPPassword

LDAPBaseDN

LDAPUserPrefix

Beispiel

Beispiel

Wichtig! Stellen Sie sicher, dass Sie das LDAP-Attribut dem Namen der neuen Organisationseinheit hinzufügen. Organisationseinheiten verwenden normalerweise das Attribut "ou" als Teil des RDN (*relativer Distinguished Name*) des Namens.

Operator-Fehler

Dieser Operator wird in den folgenden Fällen fehlschlagen:

- Der Name der neuen Organisationseinheit wird schon verwendet.
- Einige der obligatorischen Attribute, die zum Erstellen der neuen Organisationseinheit erforderlich sind, fehlen.
- Der Pfad, unter dem die Organisationseinheit erstellt werden soll, ist ungültig.
- Unfähig, eine Verbindung mit dem LDAP-Server herzustellen.

Operator "Anwender erstellen"

Der Operator "Anwender erstellen" erstellt ein LDAP-Objekt vom Typ "Anwenderkonto" auf dem LDAP-Server.

Eingabeparameter

Pfad des Anwenderkontos

Gibt den Distinguished Name des Objekts an, unter dem Sie das neue Anwenderkonto erstellen wollen.

Vorname

Gibt den Vornamen des Anwenders an.

Initialen des zweiten Vornamens

Gibt die mittleren Initialen des Anwenders an. Beachten Sie, dass Active Directory keine mittleren Initialen zulässt, die länger als sechs Zeichen sind.

Nachname

Gibt den Nachnamen des Anwenders an.

Anwender-ID

Gibt die Anwender-ID des Anwenders an.

Kennwort

Stellen Sie sicher, dass Sie ein Kennwort angeben, das den in Ihrem LDAP-Server besonders für einen Active Directory-Server festgelegten Kennwortrichtlinien-Anforderungen entspricht.

Active Directory?

Aktivieren, wenn Sie das neue Anwenderkonto in einem Active Directory-Server erstellen.

Anwender-ID als Anmeldename für den Anwender erstellen

Gibt an, ob die Anwender-ID auch der Anmeldename des Anwenders ist. Erstellen Sie in diesem Fall einen Anmeldenamen für den Anwender mit dem Format "Anwender-ID@domäne", wobei "domäne" die Domäne des Active Directory repräsentiert. Dieses Feld wird nur aktiviert, wenn das Feld "Active Directory?" aktiviert ist.

Anwender aktivieren?

Wählen Sie "Ja" aus, um den neuen Anwender als aktiv festzulegen, oder "Nein", um den neuen Anwender als inaktiv festzulegen. Dieses Feld wird nur aktiviert, wenn das Feld "Active Directory?" aktiviert ist.

Kennwort läuft für Anwender ab

Wählen Sie, ob das Anwenderkennwort gemäß der Domänenrichtlinie oder nie ablaufen soll. Dieses Feld wird nur aktiviert, wenn das Feld "Active Directory?" aktiviert ist.

Wenn das Anwenderkennwort so festgelegt wird, dass es niemals abläuft, werden Sie bei der ersten Anmeldung nicht gezwungen, das Kennwort zu ändern.

Kennwortänderung bei Erstanmeldung?

Zwingt den Anwender, das Kennwort bei der ersten Anmeldung zu ändern. Diese Option gilt nur, wenn das Kennwort ablaufen kann. Beachten Sie, dass der Anwender die Kombination festlegen kann, dass das Kennwort *nicht* abläuft und eine Änderung bei Erstanmeldung erforderlich ist. In diesem Fall legt CA Process Automation das Kennwort so fest, dass es *nicht* abläuft, und ignoriert die Aufforderung zur Kennwortänderung bei Erstanmeldung. Dieses Feld wird nur aktiviert, wenn das Feld "Active Directory?" aktiviert ist.

Beachten Sie, dass Active Directory es nicht erlaubt, dass das Kennwort eines Anwenders festgelegt wird, außer wenn CA Process Automation über SSL mit dem Active Directory-Server verbunden wird. Wenn CA Process Automation für die Verbindung nicht SSL verwendet, wird das Anwenderkonto ohne Kennwort und ohne die angegebenen Kontosteuerungen erstellt (Konto aktiviert/deaktiviert, Kennwortablauf, Kennwortänderung bei Anmeldung), und der Betriebsablauf wird in CA Process Automation fehlschlagen.

Unter Hinzufügen eines SSL-Zertifikats zu CA Process Automation (siehe Seite 266) finden Sie Informationen darüber, wie Sie ein Active Directory-Zertifikat in CA Process Automation importieren können. Nachdem das Zertifikat importiert wurde, können Sie die Kategorieeigenschaften der Verzeichnisdienste ändern, um eine SSL-Verbindung mit dem Active Directory-Server herzustellen.

Ausgabeparameter

LDAPUserBaseDn

LDAPUserFirstName

LDAPUserMiddleInits

LDAPUserLastName

LDAPUserId

LDAPUserPwd

LDAPIsAD

LDAPAsUserLogon

LDAPEnableUser

LDAPPwdExpire

LDAPForcePwdChg

remoteLDAPHost

remoteLDAPPort

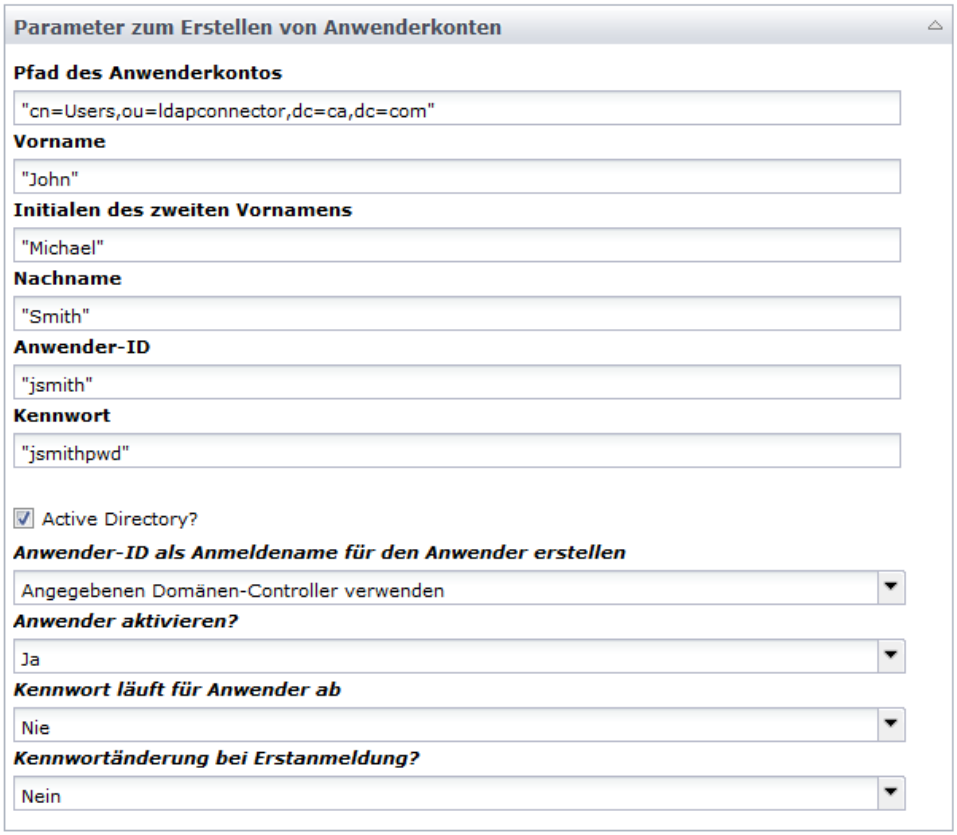
remoteLDAPUser

remoteLDAPPassword

LDAPBaseDN

LDAPUserPrefix

Beispiel



Operator-Fehler

Dieser Operator wird in den folgenden Fällen fehlschlagen:

- Der Anwender existiert schon.
- Einige der obligatorischen Attribute, die zum Erstellen eines neuen Anwenderkontos benötigt werden, fehlen.
- Der Wert eines der Attribute, mit dem das Anwenderkonto erstellt wird, ist ungültig.
- Der Pfad, unter dem das neue Anwenderkonto erstellt werden soll, ist ungültig.
- Das neue Anwenderkonto wird in einem Active Directory-Server erstellt und nicht durch SSL angeschlossen.

- Unfähig, eine Verbindung mit dem LDAP-Server herzustellen.
- Sie haben ein Anwenderkennwort angegeben, das nicht den in Ihrem LDAP-Server, besonders bei einem Active Directory, festgelegten Kennwortrichtlinien-Anforderungen entspricht. In diesem Fall gibt Active Directory eine generische Fehlermeldung zurück: WILL_NOT_PERFORM, um anzuzeigen, dass es die Operation nicht ausführen kann.

Operator "Objekt löschen"

Der Operator "Objekt löschen" löscht ein einzelnes LDAP-Objekt oder mehrere -Objekte vom LDAP-Server.

Eingabeparameter

Angegebenes Array-Feld für die Distinguished Names des Objekts verwenden

Wenn dieses Kontrollkästchen aktiviert ist, wird das Array mit den Distinguished Names für diese Anfrage verwendet.

Array mit Distinguished Names von Objekten

Gibt das Array mit den DNs der Objekte an, die gelöscht werden sollen. Dieses Feld wird nur aktiviert, wenn das Feld "Angegebenes Array-Feld für die Distinguished Names des Objekts verwenden" aktiviert ist.

Eindeutige Objektnamen

Gibt die DNs der Objekte an, die gelöscht werden sollen. Dieses Fenster wird nur aktiviert, wenn das Feld "Angegebenes Array-Feld für die Distinguished Names des Objekts verwenden" deaktiviert ist.

Umfang der Objektlöschung

Wählen Sie eine der folgenden Optionen aus:

Objekt löschen (Fehlermeldung bei vorhandenen untergeordneten Objekten)

Hiermit können Sie versuchen, jedes Objekt (von der Liste der Objekte im Array mit den Distinguished Names der Objekte oder von den Feldern mit den Distinguished Names der Objekte) so zu löschen, als ob es im LDAP-Baum unter sich keinen Unterbaum hat. Wenn ein Unterbaum für ein Objekt in der Liste existiert, schlägt die Operation in CA Process Automation fehl. Es werden aber weiterhin alle anderen Objekte in der Liste der zu löschenden Objekte gelöscht.

Objekt und Unterverzeichnis löschen (wenn vorhanden)

Hiermit können Sie versuchen, jedes Objekt (von der Liste der Objekte im Array mit den Distinguished Names der Objekte oder von den Feldern mit den Distinguished Names der Objekte) und den gesamten Unterbaum darunter zu löschen, wenn ein solcher Unterbaum vorhanden ist.

Ausgabeparameter

NumberOfObjectsToDelete

Gibt die Anzahl der Objekte an, die gelöscht werden sollen.

- Wenn "Umfang der Objektlöschung" auf "Objekt und Unterverzeichnis löschen" gesetzt ist, gibt diese Variable auch die Anzahl aller ebenfalls in den Unterbäumen gefundenen Objekte zurück.
- Wenn "Umfang der Objektlöschung" auf "Objekt löschen" gesetzt ist, gibt diese Variable die Anzahl der in der Operation festgelegten Objekte zurück.

NumberOfDeletedObjects

Enthält die Anzahl der tatsächlich gelöschten Objekte.

DeletionFailures

Gibt ein Array von Wertzuordnungen an, das nur erstellt wird, wenn die Operation fehlschlägt. In diesem Fall enthält dieses Array von Wertzuordnungen die DNs der Objekte, die nicht gelöscht wurden, zusammen mit den Fehlermeldungen, die anzeigen, warum die einzelnen Objekte nicht gelöscht wurden.

Beachten Sie, dass die Löschoperation erfolgreich ist, wenn versucht wird, ein Objekt zu löschen, das nicht im LDAP-Server existiert.

LDAPDeleteObjsUseArray

LDAPDeleteObjsArray

LDAPDeleteObjs

LDAPDeleteObjectsScope

remoteLDAPHost

remoteLDAPPort

remoteLDAPUser

remoteLDAPPassword

LDAPBaseDN

LDAPUserPrefix

Beispiele

Beispiele für eine erfolgreiche und eine fehlgeschlagene Löschung werden hier angegeben.

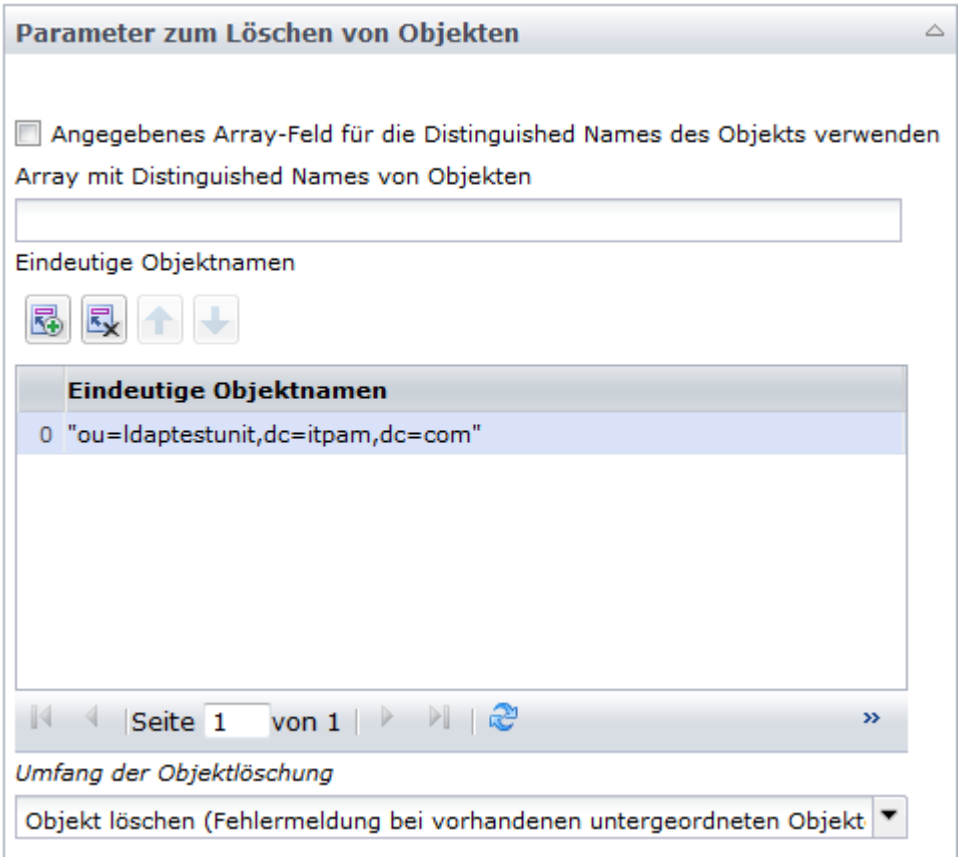
Operator-Ausfall

Dieser Operator wird in den folgenden Fällen fehlschlagen:

- Der Operator konnte keines der im Vorgang eingegebenen Objekte löschen.
- Unfähig, eine Verbindung mit dem LDAP-Server herzustellen.

Fehlgeschlagene Löschung

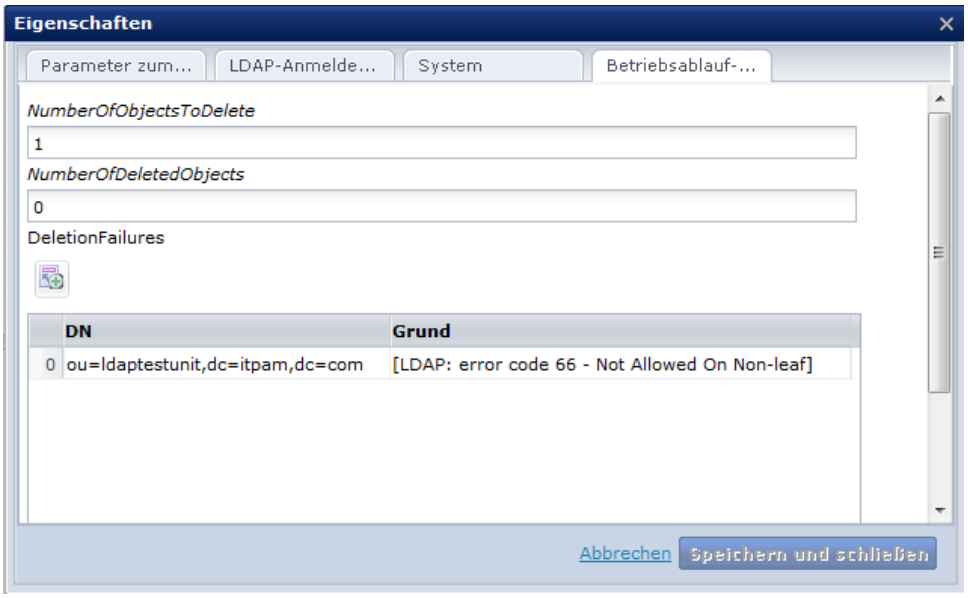
Löschen Sie ein einzelnes Objekt, und legen Sie fest, dass die Operation fehlschlägt, wenn untergeordnete Objekte (Unterbaum) unter diesem Objekt existieren.


Parameter zum Löschen von Objekten
Angegebenes Array-Feld für die Distinguished Names des Objekts verwenden
Array mit Distinguished Names von Objekten
Eindeutige Objektnamen
Eindeutige Objektnamen
0 "ou=ldaptestunit,dc=itpam,dc=com"
Seite 1 von 1
Umfang der Objektlöschung
Objekt löschen (Fehlermeldung bei vorhandenen untergeordneten Objekt

Dies ist der Operatordatensatz einer Laufzeitinstanz:

DeletionFailures	[1]
Element Type	
[0]	
Row0	
DN	ou=ldaptestunit,dc=itpam,dc=com
Reason	[LDAP: error code 66 - Not Allowed On Non-leaf]

Nachdem der Operator ausgeführt wurde, schlägt er fehl, und die folgenden Daten werden auf der Registerkarte "Betriebsablauf-Ergebnisse" angezeigt.



NumberOfObjectsToDelete

1 (da wir nur versucht haben, ein Objekt zu löschen).

NumberOfDeletedObjects

0 (da wir das Objekt nicht löschen konnten).

DeletionFailures

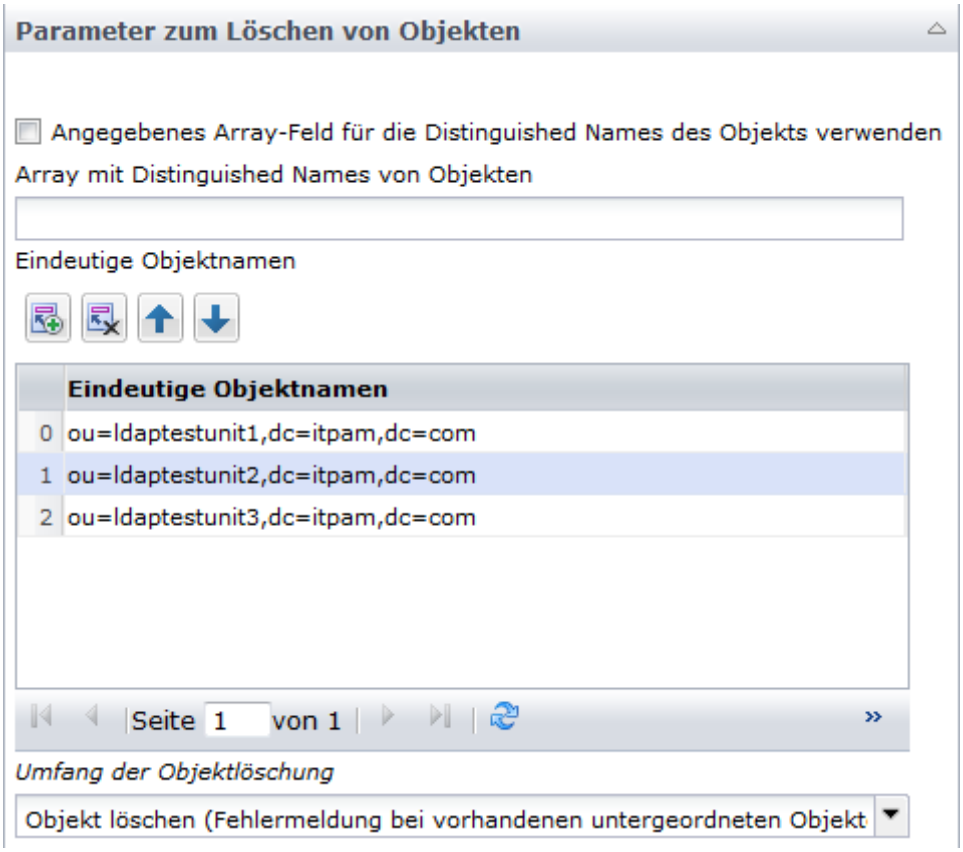
Gibt ein Array von Wertzuordnungen (ValueMaps) mit einem einzelnen Objekt an, das die DNs des Objekts enthält, das Sie zu löschen versucht haben, sowie die LDAP-Fehlermeldung, in der angegeben wird, warum nicht gelöscht wurde.

DeletionFailures	[1]
Element Type	
[0]	
Row0	
DN	ou=ldaptestunit,dc=itpam,dc=com
Reason	[LDAP: error code 66 - Not Allowed On Non-leaf]

Die LDAP-Meldung zeigt an, dass dieses Objekt unter ein Unterverzeichnis verfügt (es handelt sich um ein Nicht-Blatt).

Eine weitere fehlgeschlagene Löschung

Nehmen Sie an, dass Sie drei Objekte im Rahmen des gleichen Vorgangs löschen möchten, von denen zwei Objekte keine Unterverzeichnisse in der LDAP-Struktur unter sich haben und ein Objekt ein Unterverzeichnis unter sich hat.


Parameter zum Löschen von Objekten
Angegebenes Array-Feld für die Distinguished Names des Objekts verwenden
Array mit Distinguished Names von Objekten
Eindeutige Objektnamen
Eindeutige Objektnamen
0 ou=ldaptestunit1,dc=itpam,dc=com
1 ou=ldaptestunit2,dc=itpam,dc=com
2 ou=ldaptestunit3,dc=itpam,dc=com
Seite 1 von 1
Umfang der Objektlöschung
Objekt löschen (Fehlermeldung bei vorhandenen untergeordneten Objekt

In diesem Beispiel haben das 1. und das 2. Objekt keine Unterverzeichnisse unter sich. Das 3. Objekt in der Liste hat ein Unterverzeichnis unter sich.

Beachten Sie, dass Sie angegeben haben, ein Objekt zu löschen (und eine Fehlermeldung bei vorhandenen untergeordneten Objekten zurückgegeben wird).

Nachdem der Vorgang ausgeführt wurde, er fehlschlug und die Seite "Betriebsablauf-Ergebnisse" die folgenden Daten enthält:

NumberOfObjectsToDelete

3 (da wir versucht haben, 3 Objekte zu löschen)

NumberOfDeletedObjects

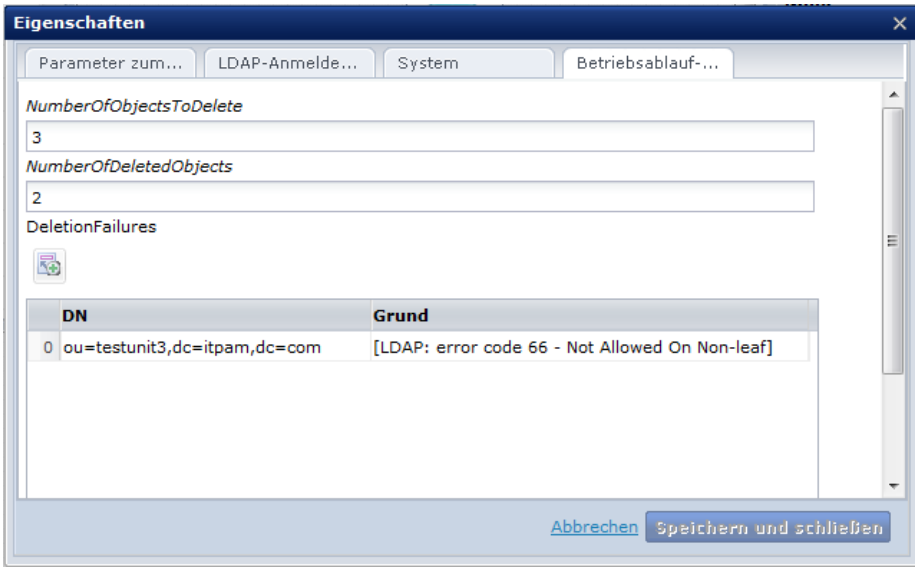
2 (da wir nur zwei Objekte löschen konnten)

DeletionFailures

Array von ValueMaps mit einem einzelnen Objekt, das Folgendes enthält:

- Den DN des Objekts, das nicht gelöscht werden konnte.
- Die LDAP-Fehlermeldung, die den Grund für die nicht erfolgte Löschung angibt.

Operation Results	
NumberOfObjectsToDelete	3
NumberOfDeletedObjects	2
DeletionFailures	[1]
Element Type	
[0]	
Row0	
DN	ou=testunit3,dc=itpam,dc=com
Reason	[LDAP: error code 66 - Not Allowed On Non-leaf]

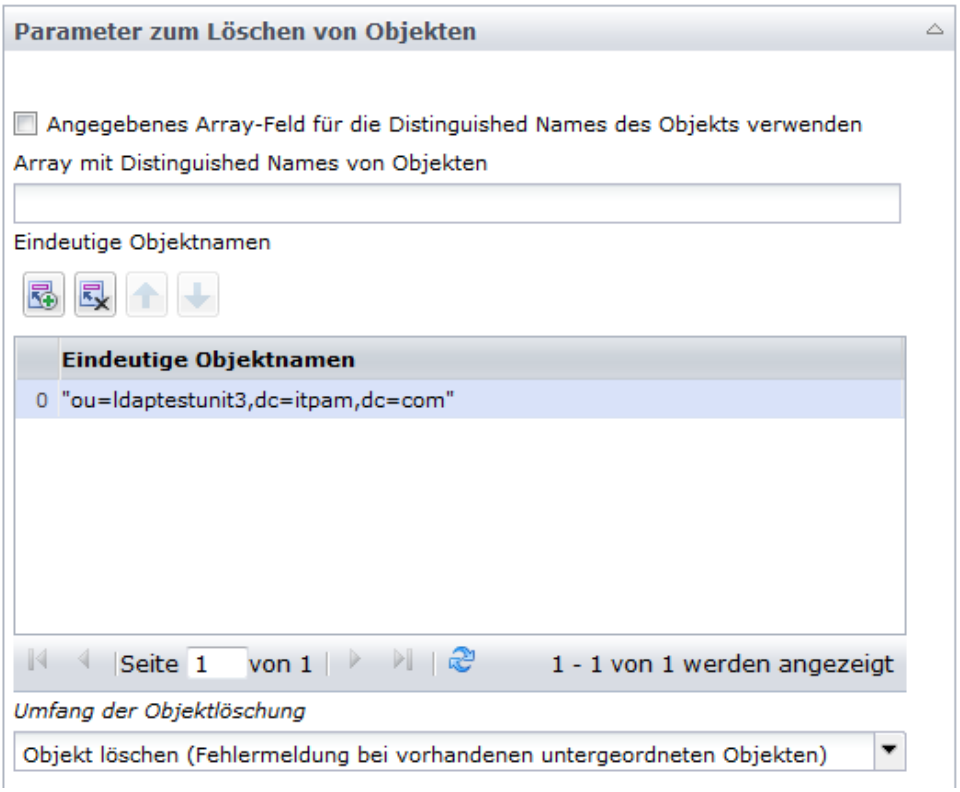


Die LDAP-Meldung zeigt an, dass dieses Objekt unter ein Unterverzeichnis verfügt (es handelt sich um ein Nicht-Blatt).

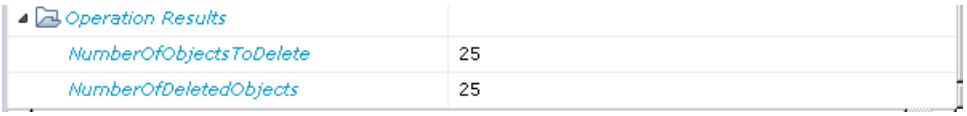
Denken Sie daran, dass der Löschvorgang die ganze Liste der zu löschenden Objekte durchsucht. Wenn ein Objekt nicht gelöscht wird, fährt der Vorgang fort, alle anderen Objekte in der Liste zu löschen, schlägt aber auch fehl, wenn er abgeschlossen ist.

Operator-Erfolg

Versuchen Sie, das gleiche Objekt und all seine untergeordneten Objekte zu löschen (mit Unterbaum darunter):

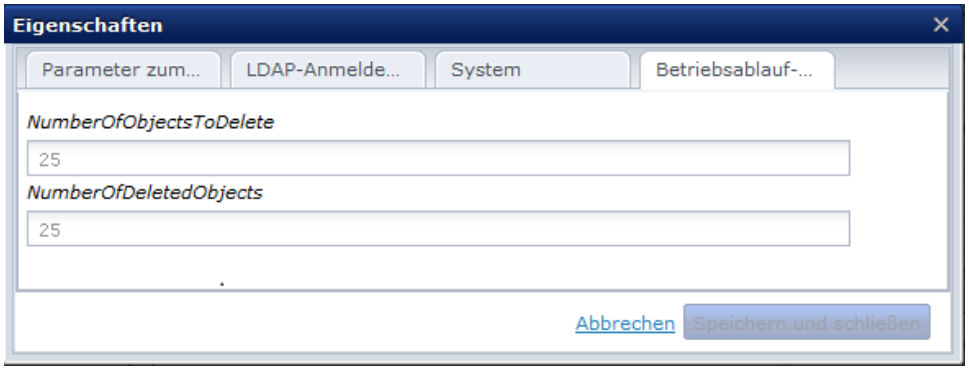

Parameter zum Löschen von Objekten
Angegebenes Array-Feld für die Distinguished Names des Objekts verwenden
Array mit Distinguished Names von Objekten
Eindeutige Objektnamen
Eindeutige Objektnamen
0 "ou=ldaptestunit3,dc=itpam,dc=com"
Seite 1 von 1
1 - 1 von 1 werden angezeigt
Umfang der Objektlöschung
Objekt löschen (Fehlermeldung bei vorhandenen untergeordneten Objekten)

Nachdem der Vorgang ausgeführt wurde, wird unter "Betriebsablauf-Ergebnisse" Folgendes angezeigt:



NumberOfObjectsToDelete: 25 (da 24 untergeordnete Objekte im Unterverzeichnis unter dem Objekt vorhanden waren).

NumberOfDeletedObjects: 25 (da das Objekt und alle untergeordneten Objekte im Unterverzeichnis darunter gelöscht werden konnten).



Beachten Sie, dass die Variable "DeletionFailures" nicht erstellt wurde.

Wenn Sie in den vorigen Beispielen versucht hatten, mehr als ein Objekt zu löschen, repräsentieren NumberOfObjectsToDelete und NumberOfDeletedObjects die Summe aller "gelöschten Objekte" und aller "zu löschenden Objekte" für alle Objekte, die bei dem Vorgang eingegeben wurden (einschließlich all ihrer Unterverzeichnisse, wenn anwendbar).

Operator "Domänencontroller abrufen"

Der Operator "Domänencontroller abrufen" ruft alle Domänencontroller vom Active Directory-Server ab. Dieser Operator gilt nur für Active Directory.

Eingabeparameter

Der Operator "Domänencontroller abrufen" enthält keine Eingabeparameter. Der Operator ruft einfach die Active Directory-Serverinformationen von der zur Operation gehörigen Seite "LDAP-Anmeldeparameter" oder aus den auf Ebene der Verzeichnisdienste-Kategorie festgelegten standardmäßigen LDAP-Anmeldeinformationen ab.

Ausgabeparameter

DomainControllers

Gibt ein Array an Zeichenfolgen (indizierte Zeichenfolge) an, das alle Domänen-Controller enthält, die vom Active Directory-Server abgerufen wurden. Diese Variable wird erstellt, wenn der Betriebsablauf erfolgreich ist.

remoteLDAPHost

remoteLDAPPort

remoteLDAPUser

remoteLDAPPassword

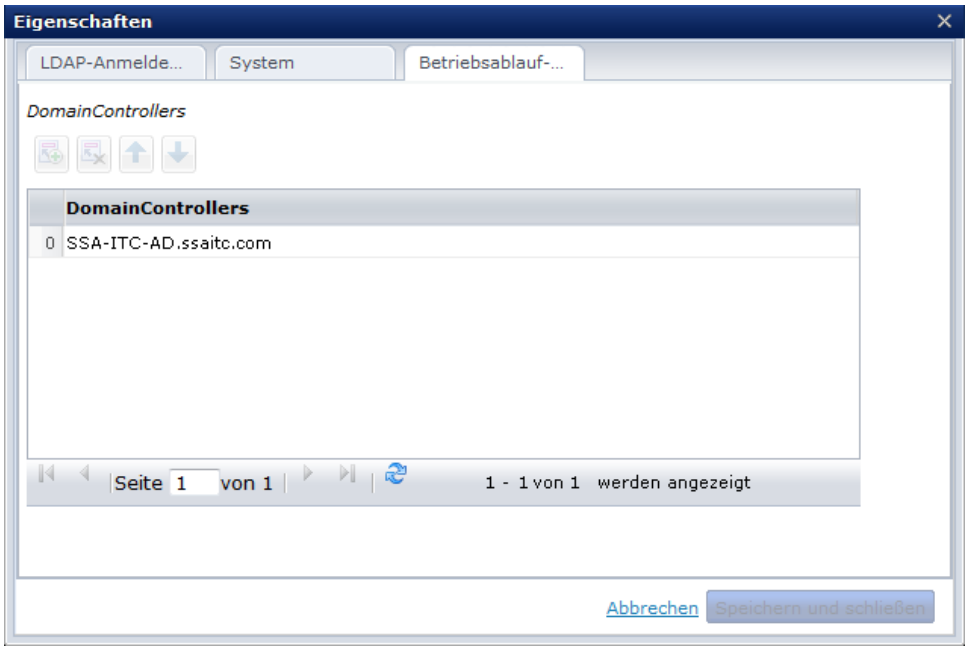
LDAPBaseDN

LDAPUserPrefix

Beispiel

Dieses Beispiel zeigt eine erfolgreiche Instanz des Operators, der DomainControllers von einem Active Directory-Server abruft. Der Operator ist erfolgreich, und die Variable "DomainControllers" wird mit den folgenden Daten erstellt:

Operation Results	
DomainControllers	[1]
[0]	SSA-ITC-AD.ssaitc.com



In diesem Beispiel haben wir jetzt einen einzelnen Domänen-Controller in dieser Domäne.

Operator-Fehler

Dieser Operator wird in den folgenden Fällen fehlschlagen:

- CA Process Automation ist nicht in der Lage, "configurationNamingContext" aus dem Active Directory-Server abzurufen.
- Es gibt keine Objekte des Typs "TDSDSA" im Active Directory-Server.
- Unfähig, eine Verbindung mit dem LDAP-Server herzustellen.

Operator "Ruhendes Konto abrufen"

Verwenden Sie den Operator "Ruhendes Konto abrufen", um alle ruhenden Konten aus Active Directory abzurufen. Sie können ein Datum und eine Anzahl von Tagen als einen ruhenden Zeitraum angeben, ein Anwenderkonto, dessen Datum der letzten Anmeldung vor diesem Zeitraum liegt, wird als ruhend bezeichnet. Dieser Operator gilt nur für Active Directory.

Eingabeparameter

Tage im Ruhezustand

Gibt die Anzahl von Tagen im Ruhezustand an.

Datum

Gibt das bestimmte Datum für ruhende Konten an. Dieses Feld wird aktiviert, wenn "Berechnetes Datum verwenden?" deaktiviert ist. Klicken Sie auf das Kalendersymbol, um ein Datum auszuwählen.

Berechnetes Datum verwenden?

Gibt an, dass der Anwender im Feld "Berechnetes Datum" eine Datensatzvariable bereitstellen kann, die einen Datumswert enthält.

Berechnetes Datum

Gibt das berechnete Datum für ruhende Konten an. Dieses Feld wird aktiviert, wenn "Berechnetes Datum verwenden?" aktiviert ist.

Letztes Anmeldeattribut

LastLogonTimeStamp

Wählen Sie dieses Feld aus, wenn ruhende Konten aus Active Directory 2003/2008 (NICHT Active Directory 2000) abgerufen werden.

Das LastLogonTimeStamp-Attribut enthält das Datum der letzten Anmeldung eines Anwenders, wird aber erst nach der im msDS-LogonTimeSyncInterval-Attribut von Active Directory angegebenen Frist unter allen Domänen-Controllern repliziert.

Wenn das Datum im Ruhezustand vor "heutiges Datum - msDS-LogonTimeSyncInterval" liegt, dann werden mittels des von einem einzelnen Domänen-Controllers abgerufenen LastLogonTimeStamp alle ruhenden Konten abgerufen.

Das msDS-LogonTimeSyncInterval-Attribut gibt die Häufigkeit (in Tagen) an, mit der die letzte Anmeldezeit für einen Anwender/Computer, die im lastLogonTimestamp-Attribut verzeichnet ist, auf alle Domänen-Controller in einer Domäne repliziert wird.

Wenn LastLogonTimeStamp verwendet wird, ruft CA Process Automation nur von einem Domänen-Controller die LastLogonTimeStamp-Informationen für jeden Anwender ab und verwendet sie, um zu entscheiden, ob das Anwenderkonto im Ruhezustand ist oder nicht.

LastLogon

Wählen Sie dieses Feld aus, wenn ruhende Konten aus Active Directory 2000, 2003 und 2008 abgerufen werden.

Das Attribut "LastLogon" enthält das Datum der letzten Anmeldung eines Anwenders, aber es wird NICHT auf allen Domänen-Controllern reproduziert.

In diesem Fall ruft CA Process Automation die Liste von allen Domänen-Controllern ab und überprüft sie, um den letzten Wert des Attributs "LastLogon" für jeden Anwender zu speichern. Schließlich wird der letzte "LastLogon"-Wert für jeden Anwender verwendet, um die ruhenden Konten zu bestimmen.

Ausgabeparameter

DormantAccounts

Gibt ein Array von Wertzuordnungen an, wobei jede Wertzuordnung ein ruhendes Konto repräsentiert. Jede Wertzuordnung enthält die folgenden Schlüssel/Werte:

DN

Gibt den Distinguished Name des ruhenden Anwenderkontos an.

Last Logon Date

Gibt das Datum der letzten Anmeldung des ruhenden Anwenderkontos an.

Dormant Days

Gibt die Anzahl von Tagen zwischen dem Datum der letzten Anmeldung des ruhenden Anwenderkontos und dem in der Operation festgelegten Tag im Ruhezustand an. Beachten Sie, dass diese Zahl gerundet wird, zum Beispiel, wenn das Datum der letzten Anmeldung eines Anwenders 15.05.2009 12:25:49 PM ist und ein ruhendes Datum am 18.05.2009 vorliegt, dann beträgt die Anzahl von Tagen im Ruhezustand 4 (nicht 3).

Die Werte "Last Logon Date" und "Dormant Days" werden beide für jedes ruhende Anwenderkonto, bei dem sich noch niemals jemand angemeldet hat, auf -1 festgelegt.

NoLogonParameterAccounts

Gibt ein Array an Zeichenfolgen (indizierte Zeichenfolge) an, das die DNs der Anwenderkonten enthält, deren Attribute "LastLogonTimeStamp" oder "LastLogon" nicht im Active Directory festgelegt wurden (je nachdem, welches Attribut vom Anwender für die Suche ausgewählt wurde). Wenn für alle Anwenderkonten das ausgewählte Attribut festgelegt wurde, dann ist diese Variable leer.

LDAPADDormantDays

LDAPADDormantDate

LDAPADUseCalculatedDate

LDAPADCalculatedDate

LDAPADLastLogonAttr

remoteLDAPHost

remoteLDAPPort

remoteLDAPUser

remoteLDAPPassword

LDAPBaseDN

LDAPUserPrefix

Operator "Objekt abrufen"

Der Operator "Objekt abrufen" ruft LDAP-Objekte eines beliebigen Typs vom LDAP-Server ab. Sie können den Suchpfad, den Suchfilter (siehe Seite 236), den Suchumfang, die Attribute (siehe Seite 237), die mit jedem Objekt abgerufen werden sollen, und die Sortierkriterien angeben.

Eingabeparameter

Kriterien abrufen

Suchpfad-Anfang

Gibt das Startverzeichnis für die Suche an, z. B. CN=Users,dc=domainpart,dc=company-name,dc=top-level-domain-name.

Abzurufender Umfang

Wählen Sie eine der folgenden Optionen:

Gesamtes Unterverzeichnis

Es wird der ganze Unterbaum durchsucht (einschließlich des Objekts im Suchpfad).

Eine Ebene

Es werden nur die Objekte durchsucht, die sich im Suchpfad direkt unter dem Objekt befinden.

Objekt

Es wird nur das Objekt im Suchpfad durchsucht.

Ergebnisbegrenzung

Größtmögliche Anzahl von Einträgen, die zurückgegeben werden.

- Wenn Sie 0 oder nichts in dieses Feld eingeben, verwendet CA Process Automation den Wert, der im Feld "Höchstanzahl von Suchergebnissen" festgelegt ist (aus der Konfiguration der Verzeichnisdienste-Kategorie).
- Wenn Sie einen Wert in dieses Feld eingeben, verwendet CA Process Automation entweder den Wert dieses Feldes oder den Wert aus dem Feld "Höchstanzahl von Suchergebnissen" (aus der Konfiguration der Verzeichnisdienste-Kategorie), je nachdem, welcher Wert kleiner ist.

Zeitlimit für Abruf

Zeit (in Sekunden), die gewartet werden soll, bis eine Zeitüberschreitung für die Suche auftritt. Wenn dieses Limit 0 ist (oder nichts eingegeben wurde), wird kein Zeitlimit für die Suche festgelegt.

Objekttyp abrufen

Wählen Sie eine der folgenden Optionen:

- Anwender
- Computer
- Gruppe
- Organisationseinheit
- Rolle
- Sonstige

Mit diesem Feld wird der Filterwert gesteuert, der im Feld "Filter abrufen" angezeigt wird.

Filter abrufen

Wählen Sie den für diese Suche zu verwendenden Suchfilter aus.

Sie können:

Verwenden eines generischen Filters, indem der Anwender im Feld "Objekttyp abrufen" der Eintrag "Anwender", "Gruppe", "Organisationseinheit", "Rolle" oder "Computer" auswählt. Danach wird der zugeordnete Filterwert im Feld "Filter abrufen" angezeigt und dieses Feld in ein schreibgeschütztes Feld geändert.

oder

Eingeben eines eigenen Filters, indem Sie im Feld "Objekttyp abrufen" den Eintrag "Andere" auswählen. Dadurch wird "objectclass=" im Feld "Filter abrufen" angezeigt und dieses Feld schreibbar gemacht, sodass Sie ihren eigenen Filterwert eingeben können. Der Ausdruck muss syntaktisch richtig sein, denn sonst kann die Suche fehlschlagen. Eine Einführung in die Syntax von LDAP-Suchfiltern finden Sie unter Grundlagen zu LDAP-Suchfiltern (siehe Seite 236).

Sie können unterschiedliche Filter für "Anwender", "Gruppe", "Organisationseinheit", "Rolle" oder "Computer" verwenden, indem Sie im Feld "Objekttyp abrufen" den Eintrag "Andere" auswählen und einen Filterwert in das Feld "Filter abrufen" eingeben. Die bereitgestellten generischen Suchfilter funktionieren mit einigen LDAP-Servern möglicherweise nicht. Dies ist insbesondere der Fall, wenn der LDAP-Server einige der Objektklassen nicht unterstützt, die in den Filtern aufgelistet sind.

Als Array-Variable festgelegte Attribute abrufen?

Ist diese Option aktiviert, können Sie eine Datensatzvariable bereitstellen, die ein Array von abzurufenden Attributen enthält.

Variable aus dem Attribute-Array abrufen

Die Datensatzvariable, die ein Array von abzurufenden Attributen bereitstellt. Dieses Feld ist aktiviert, wenn das Feld "Als Array-Variable festgelegte Attribute abrufen?" aktiviert ist.

Attributliste abrufen

Gibt eine Liste von für diesen Suchfilter abzurufenden Attributen an. Diese Liste ist aktiviert, wenn das Feld "Als Array-Variable festgelegte Attribute abrufen?" nicht aktiviert ist.

Sortierungskriterien

Als Array-Variable festgelegte Felder sortieren?

Ist diese Option aktiviert, können Sie eine Datensatzvariable bereitstellen, die ein Array von Attributen enthält, die zum Sortieren der abgerufenen Daten verwendet werden.

Array-Variable für die Sortierung von Listenfeldern

Die Datensatzvariable, die ein Array von Attributen bereitstellt, die für die Sortierreihenfolge verwendet werden sollen. Dieses Feld ist aktiviert, wenn das Feld "Als Array-Variable festgelegte Felder sortieren?" aktiviert ist.

Liste für Sortierreihenfolge

Liste der Attribute, die für die Sortierreihenfolge verwendet werden sollen. Diese Liste ist aktiviert, wenn das Feld "Als Array-Variable festgelegte Felder sortieren?" nicht aktiviert ist.

Wenn nichts in den Abschnitt "Sortierungskriterien" eingegeben wird, werden die abgerufenen Objekte nicht sortiert.

Einige LDAP-Server (zum Beispiel OpenLdap) unterstützen das "Sortieren" von Daten nicht. In diesem Fall könnte der Operator aus folgendem Grund fehlschlagen: [LDAP: Fehlercode 12 - kritische Erweiterung wird nicht erkannt]. Geben Sie in diesem Fall keine Sortierkriterien an. Dies ist eine Beschränkung im LDAP-Server, nicht in CA Process Automation.

Grundlagen zu LDAP-Suchfiltern

Die LDAP-Suchfiltersyntax ist ein logischer Ausdruck in Präfixnotation, wobei der logische Operator vor den zugeordneten Argumenten angezeigt wird.

Beispiel: (&(givenname=John)(sn=Green))

Im Filter oben ist & der UND-Operator, und er wird vor seinen Argumenten angezeigt. In diesem Beispiel suchen wir nach LDAP-Objekten mit John als givenname (givenname ist das LDAP-Attribut für Vornamen) und Green als sn (sn ist das LDAP-Attribut für Nachnamen).

Jedes Element im Filter wird mithilfe eines LDAP-Attributbezeichners und entweder eines Attributwerts oder Symbolen, die den Attributwert anzeigen, gebildet. Jedes Element muss zudem in Klammern gesetzt werden, wie in "(sn=Green)".

Elemente innerhalb eines Filters werden mithilfe logischer Operatoren kombiniert, um logische Ausdrücke zu erstellen. Jeder logische Ausdruck kann wiederum mit anderen Elementen, die logische Ausdrücke sind, kombiniert werden. Dies ist bei einigen in CA Process Automation verwendeten Filtern so:

```
(&(|(objectclass=user)(objectclass=person))(!(objectclass=computer)))
```

In diesem Filter wird nach allen Objekten gesucht, bei denen "objectclass" den Wert "user" ODER "person" enthält:

```
(|(objectclass=user)(objectclass=person))
```

UND "objectclass" ist *nicht* gleich "computer"

```
(!(objectclass=computer))
```

Beachten Sie das Symbol & am Anfang des Filters, mit dem diese beiden Segmente zu einem logischen UND verbunden werden.

Beachten Sie, dass das LDAP-Attribut "objectclass" den Typ eines LDAP-Objekts im LDAP-Verzeichnis speichert.

Einige logische Operatoren, die zum Erstellen von Filtern verwendet werden, sind in der folgenden Tabelle aufgelistet:

Symbol	Beschreibung
=	Gleichheit Beispiel: (givenname=John) Nach Objekten mit John als Vorname suchen.

&	Logisches UND Beispiel: (&(givenname=John)(sn=Green)) Nach Objekten mit John als Vorname und Green als Nachname suchen
|	Logisches ODER Beispiel: (|(givenname=John)(givenname=Michael)) Nach Objekten mit John oder Michael als Vorname
!	Logisches NICHT Beispiel: (&(givenname=John)(!(sn=Green))) Nach Objekten mit John als Vorname und nicht Green als Nachname suchen
>=	Größer als Beispiel: (numsubordinates>=2) Nach Objekten mit 2 oder mehr untergeordneten Knoten in der LDAP-Struktur suchen.
<=	kleiner als Beispiel: (numsubordinates<=2) Nach Objekten mit 2 oder weniger untergeordneten Knoten in der LDAP-Struktur suchen.
=*	Vorhandensein Das Objekt muss über das Attribut verfügen, sonst ist sein Wert irrelevant. Beispiel: (givenname=*) Nach Objekten mit dem Attribut "givenname" suchen.
*	Platzhalter Beispiel: (givenname=Joh*) Nach Objekten suchen, deren Attribut "givenname" mit "Joh" anfängt

Allgemeine LDAP-Attributnamen

Einige übliche LDAP-Attribute werden unten aufgelistet. Die vollständige Liste von LDAP-Objektklassen und im LDAP-Server-Schema verwendeten Attributen befindet sich auf dem LDAP-Server.

LDAP-Attributname	Beschreibung
cn	Allgemeines Namensattribut, das den Namen des Objekts enthält
dc	Domänenkomponenten-Attribut
objectClass	Objektklassen-Attribut, das den/die LDAP-Typ(en) des Objekts enthält
distinguishedName	Distinguished-Name-Attribut in Active Directory Dies ist das Attribut, das Objekt in Active Directory eindeutig identifiziert.

LDAP-Attributname	Beschreibung
entrydn	Distinguished-Name-Attribut in LDAP-Servern (nicht Active Directory) Dies ist das Attribut, das Objekt auf einem LDAP-Server eindeutig identifiziert.
o	Allgemeines Namensattribut, das den Namen des Objekts enthält
ou	Organisationseinheit-Namensattribut, das den Namen der Organisationseinheit enthält
sn	Nachnamen-Attribut, das den Familiennamen einer Person enthält
givenName	Vornamens-Attribut, das den Vornamen einer Person enthält
personalTitle	Personentitel-Attribut, das den Titel einer Person enthält Beispiele für Personentitel sind "Hr.", "Dr.", "Prof." und "Dipl".
Initiale	Initiale-Attribut, das die Initiale von einigen oder allen Namen einer Person, mit Ausnahm des/der Zunamen(s) enthält
uid	Anwender-ID-Attribut
userPassword	Kennwort-Attribut, das das Kennwort eines Anwenders enthält Kennwörter werden mithilfe einer Oktettzeichenfolgensyntax gespeichert und nicht verschlüsselt.
title	Titel-Attribut, das die designierte Position oder Funktion des Objekts innerhalb der Organisation angibt
mail	Mail-Attribut, das die E-Mail-Adresse eines Anwenders enthält
company	Attribut, das den Namen des Unternehmens oder der Organisation enthält
Abteilung	Attribut für den Abteilungsnamen
manager	Attribut, das den Namen des Vorgesetzten/Managers enthält
mobile	Attribut, das die Mobiltelefonnummer enthält
homephone	Attribut, das die Festnetztelefonnummer enthält
telephoneNumber	Attribut, das eine Telefonnummer enthält
facsimileTelephoneNumber	Attribut, das eine Faxnummer enthält
postalAddress	Postanschriften-Attribut, das für die physische Zustellung von postalischen Nachrichten benötigte Information enthält
postalCode	Postleitzahlen-Attribut Wenn dieser Attributwert vorhanden ist, dann bildet er einen Teil der Postanschrift des Objekts.
c	Attribut, das den Ländernamen in Form eines zwei Buchstaben langen ISO 3166-Ländercode enthält

LDAP-Attributname	Beschreibung
l	Attribut, das den Namen der Örtlichkeit, z.B. Stadt, Bezirk oder eine andere geographische Region enthält
st	Attribut,das den Namen des Bundesstaats/-lands oder der Provinz enthält
street	Straßen-Attribut, das die physische Adresse des Objekts enthält, z. B. eine Adresse für die Paketzustellung
Eigentümer	Attribut, das den Namen eines Objekts angibt, das für das zugehörige Objekt verantwortlich ist Der Wert ist ein eindeutiger Name.
description	Attribut, das eine in Textform vorliegende Beschreibung des Objekts enthält
seeAlso	Attribut "Siehe auch".
serialNumber	Attribut, das die Seriennummer eines Geräts speichert
member	Dieses Attribut wird in Einträgen zum Definieren von Gruppen verwendet Es verfügt über Distinguished-Name-Syntax, sodass jeder Wert im Grunde genommen ein Zeiger auf einen anderen Verzeichniseintrag ist. Beachten Sie, dass bei der Standardobjektklasse groupOfNames das Attribut "member" obligatorisch ist. Da Attribute keine Leerwerte enthalten können, müssen alle Gruppen stets mindestens ein Mitglied haben.
uniqueMember	Das uniqueMember-Attribut ähnelt dem oben beschriebenen member-Attribut. Es wird verwendet, um die eindeutigen Mitglieder in einem groupOfUniqueNames-Objekt zu speichern
sAMAccountName	Altes NT-4.0-Anmeldenamensattribut (nur Active Directory), das in einer Active-Directory-Domäne eindeutig sein muss
LastLogonTimeStamp	Attribut (nur Active Directory 2003/2008) mit dem Zeitstempel der letzten Anmeldung, das das Datum der letzten Anmeldung eines Anwenders enthält und erst nach der im msDS-LogonTimeSyncInterval-Attribut von Active Directory angegebenen Frist unter allen Domänen-Controllern repliziert wird.
LastLogon	Attribut (nur Active Directory), das das Datum der letzten Anmeldung eines Anwenders enthält, aber NICHT unter allen Domänen-Controllern repliziert wird

Ausgabeparameter

RetrievedObjects

Ein Array von Wertzuordnungen, wobei jede Wertzuordnung die für jedes Objekt abgerufenen Attribute enthält. Diese Variable wird nur erstellt, wenn der Vorgang erfolgreich ist.

LDAPSearchPath

LDAPSearchScope

ResultsLimit

LDAPGetTimeLimit

LDAPSearchType

LDAPGetFilter

LDAPGetAttributesType

LDAPGetAttributesArray

LDAPGetAttributes

LDAPGetSortAttributesType

LDAPGetSortAttributesArray

LDAPGetSortAttributes

remoteLDAPHost

remoteLDAPPort

remoteLDAPUser

remoteLDAPPassword

LDAPBaseDN

LDAPUserPrefix

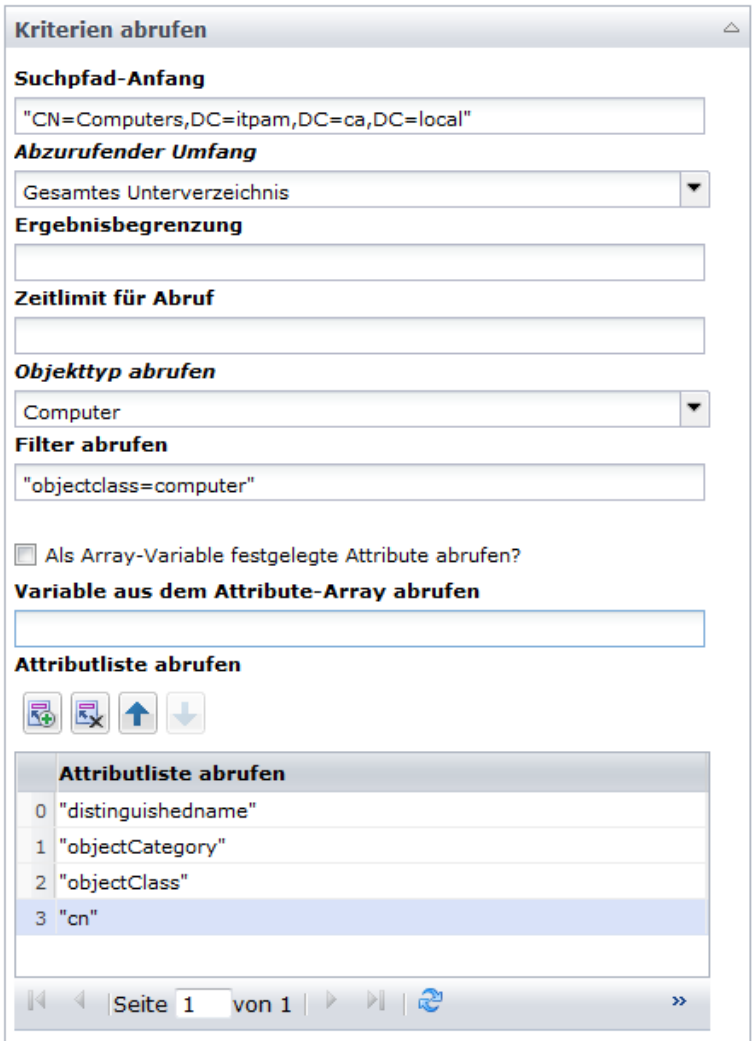
Hinweise:

- Die Attributnamen, die in die Abschnitte für Sortieren und Abrufen eingegeben werden, müssen die LDAP-Namen dieser Attribute sein, wie sie im LDAP-Serverschema angegeben sind. Um zum Beispiel das Attribut "Nachname" abzurufen, müssen Sie den LDAP-Namen dieses Attributs verwenden: "sn". Um das Attribut "Vorname" abzurufen, müssen Sie das Attribut "givenname" verwenden, und so weiter. Siehe Allgemeine LDAP-Attributnamen (siehe Seite 237).
- Die LDAP-Namen unterscheiden sich von den Attributanzeigenamen.
- Die meisten LDAP-Server unterscheiden sich in den Anzeigenamen der LDAP-Attribute, aber sie alle müssen die LDAP-Namen dieser Attribute unterstützen. Aus diesem Grund müssen die LDAP-Namen der Attribute statt der Anzeigenamen verwendet werden.

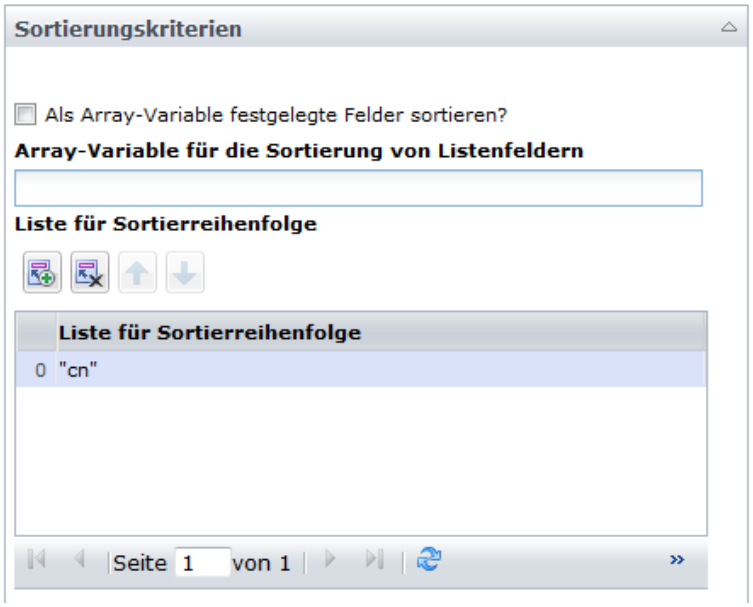
- Sie müssen die Namen der abzurufenden Attribute angeben. Andernfalls gibt CA Process Automation in der Variable RetrievedObjects keine Daten zurück.
- Wenn kein Objekt unter dem angegebenen Suchpfad gefunden wird, ist der Suchvorgang erfolgreich und die RetrievedObjects-Variable leer. Der Suchvorgang schlägt in diesen Fall nicht fehl.

Beispiele

Beispiel - Verwenden eines generischen Filters

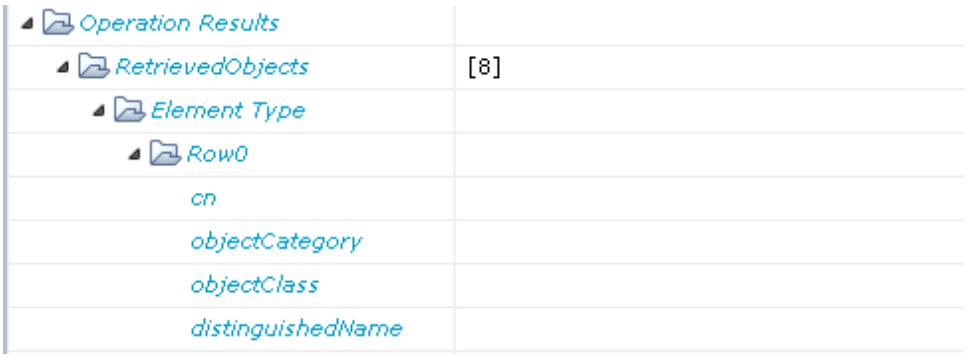


In diesem Beispiel versuchen wir, alle Computerkonten unter dem Pfad "CN=Computers,DC=itpam,DC=ca,DC=local" abzurufen. Wir bitten insbesondere um die Angabe der Attribute "cn", "distinguishedname", "objectcategory" und "objectclass" zu diesen Konten.

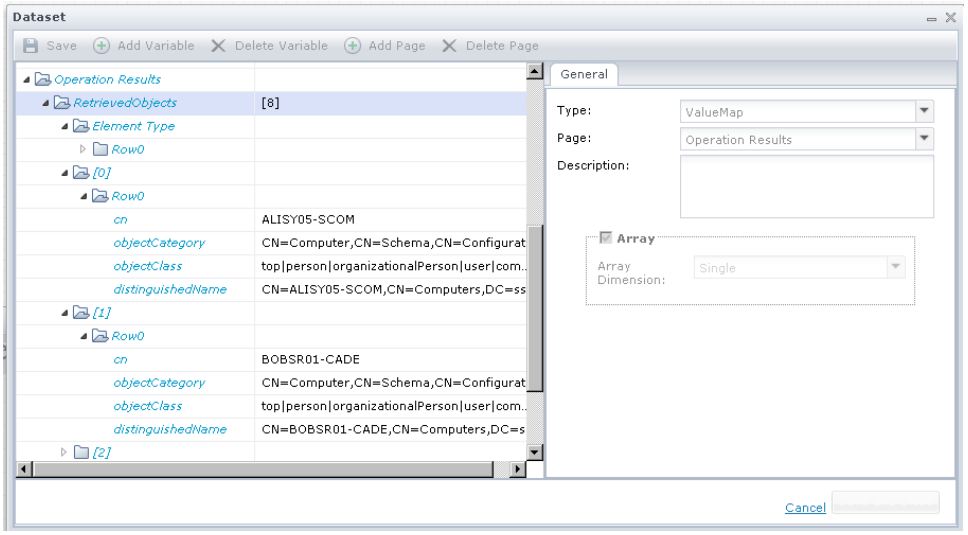


Wir sortieren auch die zurückgegebenen Anwenderkonten nach "uid".

Nachdem die Operation ausgeführt wurde, wurde sie erfolgreich abgeschlossen, und die Variable RetrievedObjects wurde folgendermaßen erstellt:



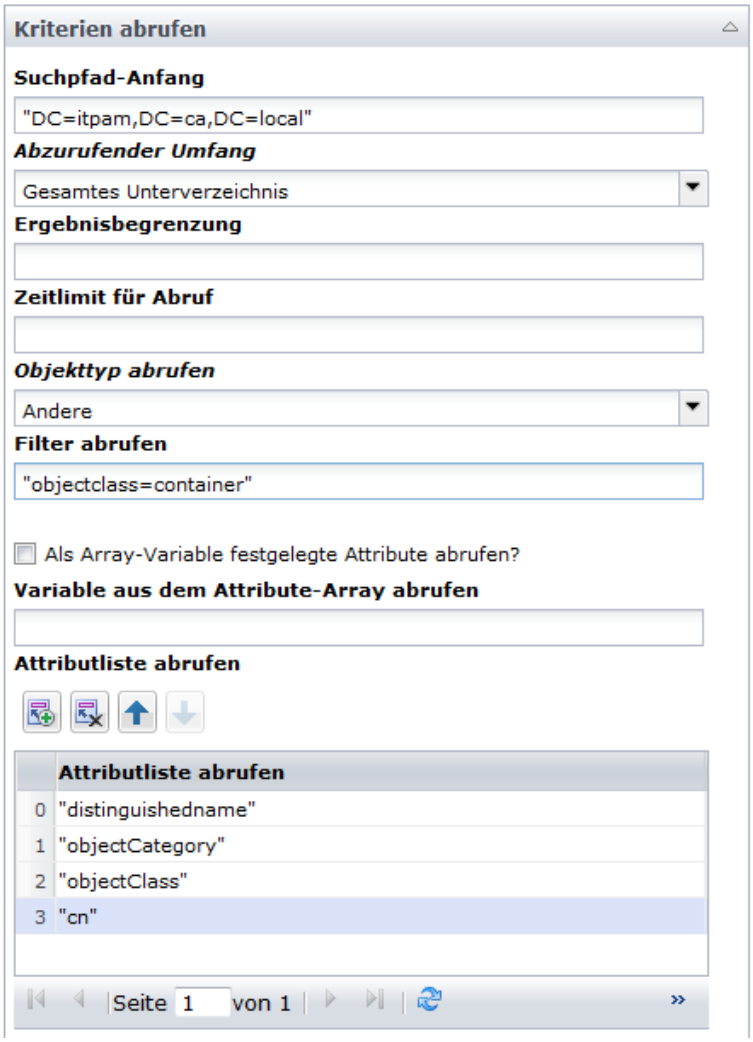
Jede Wertzuordnung in der Variable RetrievedObjects zeigt jetzt die für jedes Computerkontoattribut abgerufenen Werte. In diesem Beispiel enthält die Variable RetrievedObjects acht Objekte.



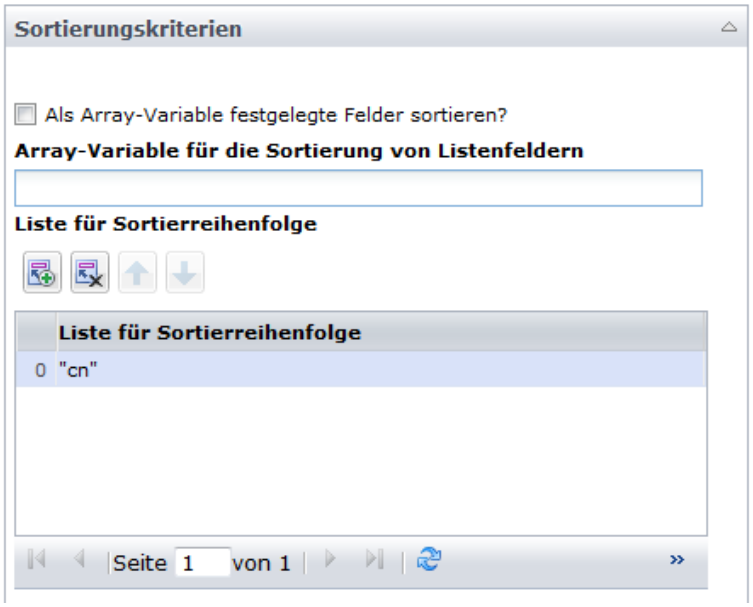
Beachten Sie zudem, dass mehrwertige Attribute (in diesem Beispiel "objectclass") auch mit einem "|" zwischen den einzelnen Werten zurückgegeben werden:

objectClass	top|person|organizationalPerson|user|computer

Beispiel - Verwenden eigener Filter



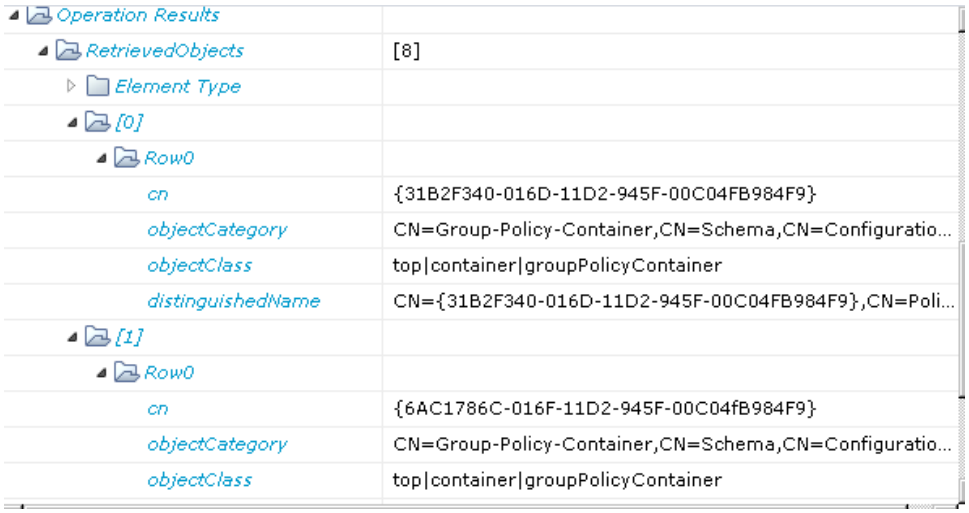
In diesem Beispiel verwenden wir unseren eigenen Filter (Hinweis: der abzurufende Objekttyp ist auf "Andere" festgelegt), um alle Containerkonten unter dem Pfad "DC=itpam,DC=ca,DC=local" abzurufen. Wir bitten insbesondere um die Angabe der Attribute "cn", "distinguishedname", "objectcategory" und "objectclass" zu diesen Konten.

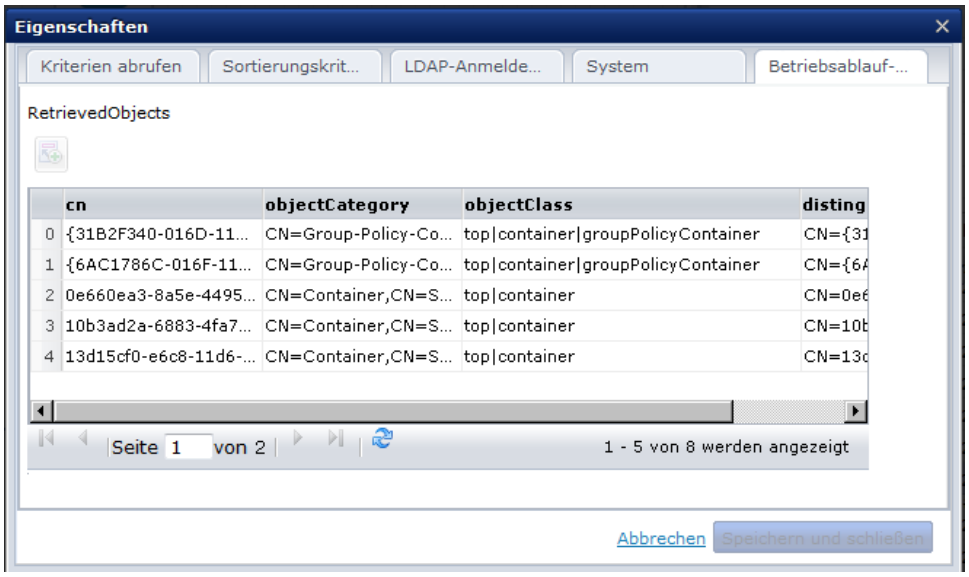


Wir sortieren dann auch die zurückgegebenen Anwenderkonten nach "cn".

Nachdem die Operation ausgeführt wurde, wurde sie erfolgreich abgeschlossen, und die Variable RetrievedObjects wurde folgendermaßen erstellt:

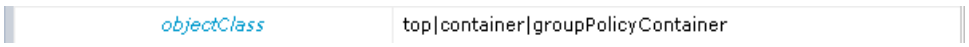
Operation Results	
RetrievedObjects	[8]
Element Type	
Row0	
cn	
objectCategory	
objectClass	
distinguishedName	





Jede Wertzuordnung in der Variable RetrievedObjects zeigt jetzt die für jedes Containerkontoattribut abgerufenen Werte. Die VariableRetrievedObjects enthält in diesem Beispiel 86 Objekte.

Beachten Sie zudem, dass mehrwertige Attribute (in diesem Beispiel "objectclass") auch mit einem "|" zwischen den einzelnen Werten zurückgegeben werden:



Operator-Fehler

Dieser Operator wird in den folgenden Fällen fehlschlagen:

- Der Suchpfad ist auf dem LDAP-Server nicht vorhanden.
- Das Zeitlimit für die Suche wurde überschritten.
- Unfähig, eine Verbindung mit dem LDAP-Server herzustellen.

Je nach LDAP-Server *kann* diese Operation in den folgenden Fällen fehlschlagen, da einige LDAP-Server diese Situation für einen Fehler halten, andere dagegen nicht:

- Der Suchfilter ist ungültig.
- Die zurückgegebenen Attribute sind ungültig.
- Die Sortierattribute sind ungültig.

Operator "Anwender abrufen"

Der Operator "Anwender abrufen" ruft LDAP-Objekte vom Typ Anwenderkonto vom LDAP-Server auf. Sie können den Suchpfad, den Suchfilter, den Suchumfang, die mit jedem Objekt abzurufenden Attribute und die Sortierkriterien angeben.

Eingabeparameter

Eingabeparameter für den Operator "Anwender abrufen" schließen Abruf- und Sortierungskriterien ein.

Kriterien abrufen

Suchpfad-Anfang

Gibt den Startpunkt für die Suche an, wie z. B. CN=Anwender,dc=Domänenteil,dc=Firmenname,dc=Top-Level-Domänenname.

Abzurufender Umfang

Wählen Sie eine der folgenden Optionen:

Gesamtes Unterverzeichnis

Durchsucht den ganzen Unterbaum (einschließlich des Objekts am Suchpfad).

Eine Ebene

Durchsucht die Objekte direkt unter dem Objekt am Suchpfad.

Objekt

Sucht das Objekt nur am Suchpfad.

Ergebnisbegrenzung

Größtmögliche Anzahl von Einträgen, die zurückgegeben werden.

- Geben Sie 0 oder nichts in dieses Feld ein, um den Wert aus dem Feld "Höchstanzahl von Suchergebnissen" (aus der Konfiguration der Verzeichnisdienste-Kategorie) zu verwenden.
- Geben Sie einen Wert in dieses Feld ein, um entweder den Wert dieses Feldes oder den Wert aus dem Feld "Höchstanzahl von Suchergebnissen" (aus der Konfiguration der Verzeichnisdienste-Kategorie) zu verwenden, je nachdem, welcher Wert kleiner ist.

Zeitlimit für Abruf

Gibt die Zeit (in Sekunden) an, die gewartet werden muss, bevor diese Suche das Zeitlimit erreicht.

Wenn 0 oder nichts eingegeben wird, wird kein Zeitlimit für die Suche festgelegt.

Filter abrufen

Gibt den für diese Suche zu verwendenden Suchfilter an.

- Der generische Suchfilter sucht nach Anwenderkonten in LDAP, und Sie können diesen Filter nach Bedarf anpassen. Dieses Feld ist beschreibbar (kann geändert werden). Eine Einführung in die Syntax von LDAP-Suchfiltern finden Sie unter Grundlagen zu LDAP-Suchfiltern (siehe Seite 236).
- Der generische Suchfilter funktioniert möglicherweise mit einigen LDAP-Servern nicht, besonders wenn der LDAP-Server einige der im Filter aufgelisteten Objektklassen (objectclass) nicht unterstützt:

  "(&(|(objectclass=user)(objectclass=person))(!(objectclass=computer)))"

- Der Ausdruck muss syntaktisch richtig sein, denn sonst kann die Suche fehlschlagen.
- Informieren Sie sich bei Ihrem LDAP-Administrator über die unterstützten LDAP-Objektklassen.

Als Array-Variable festgelegte Attribute abrufen?

Aktivieren Sie dieses Kontrollkästchen, um eine Datensatzvariable bereitzustellen, die ein Array mit abzurufenden Attributen enthält.

Variable aus dem Attribute-Array abrufen

Gibt die Datensatzvariable an, die ein Array mit abzurufenden Attributen bereitstellt. Dieses Feld ist aktiviert, wenn das Feld "Als Array-Variable festgelegte Attribute abrufen?" aktiviert ist.

Attributliste abrufen

Gibt eine Liste von für diesen Suchfilter abzurufenden Attributen an. Diese Liste ist aktiviert, wenn das Feld "Als Array-Variable festgelegte Attribute abrufen?" nicht aktiviert ist.

Sortierungskriterien

Als Array-Variable festgelegte Felder sortieren?

Aktivieren Sie dieses Kontrollkästchen, um eine Datensatzvariable bereitzustellen, die ein Array mit Attributen zum Sortieren der abgerufenen Daten enthält.

Array-Variable für die Sortierung von Listenfeldern

Gibt die Datensatzvariable an, die ein Array von Attributen bereitstellt, die für die Sortierreihenfolge verwendet werden sollen. Dieses Feld wird aktiviert, wenn das Feld "Als Array-Variable festgelegte Felder sortieren?" aktiviert ist.

Liste für Sortierreihenfolge

Liste der Attribute, die für die Sortierreihenfolge verwendet werden sollen. Dieses Feld wird aktiviert, wenn das Feld "Als Array-Variable festgelegte Felder sortieren?" deaktiviert ist.

Wenn nichts in den Abschnitt "Sortierkriterien" eingegeben wird, werden die abgerufenen Objekte nicht sortiert.

Hinweis: Einige LDAP-Server (wie OpenLdap) unterstützen das "Sortieren" von Daten nicht. In diesem Fall könnte der Vorgang aus folgendem Grund fehlschlagen: [LDAP: Fehlercode 12 - kritische Erweiterung wird nicht erkannt]. Geben Sie in diesem Fall keine Sortierkriterien an. Dies ist eine Beschränkung im LDAP-Server, nicht in CA Process Automation.

Ausgabeparameter

Anwenderkonten

Dies ist ein Array von Wertzuordnungen, wobei jede Wertzuordnung die für den Anwender abgerufenen Attribute enthält. Diese Variable wird nur erstellt, wenn der Vorgang erfolgreich ist.

LDAPSearchPath

LDAPSearchScope

ResultsLimit

LDAPGetTimeLimit

LDAPGetFilter

LDAPGetAttributesType

LDAPGetAttributesArray

LDAPGetAttributes

LDAPGetSortAttributesType

LDAPGetSortAttributesArray

LDAPGetSortAttributes

remoteLDAPHost

remoteLDAPPort

remoteLDAPUser

remoteLDAPPassword

LDAPBaseDN

LDAPUserPrefix

Attributnamen, die in die Abschnitte zum Sortieren und Abrufen eingegeben wurden, müssen die LDAP-Namen dieser Attribute sein, wie sie im LDAP-Server-Schema angegeben sind. Um zum Beispiel das Attribut "Nachname" abzurufen, müssen Sie den LDAP-Namen dieses Attributs verwenden: "sn". Um das Attribut "Vorname" abzurufen, müssen Sie das Attribut "givenname" verwenden, und so weiter. Weitere Informationen finden Sie im Thema "Allgemeine LDAP-Attributnamen (siehe Seite 237)".

LDAP-Namen unterscheiden sich von den Anzeigenamen der Attribute.

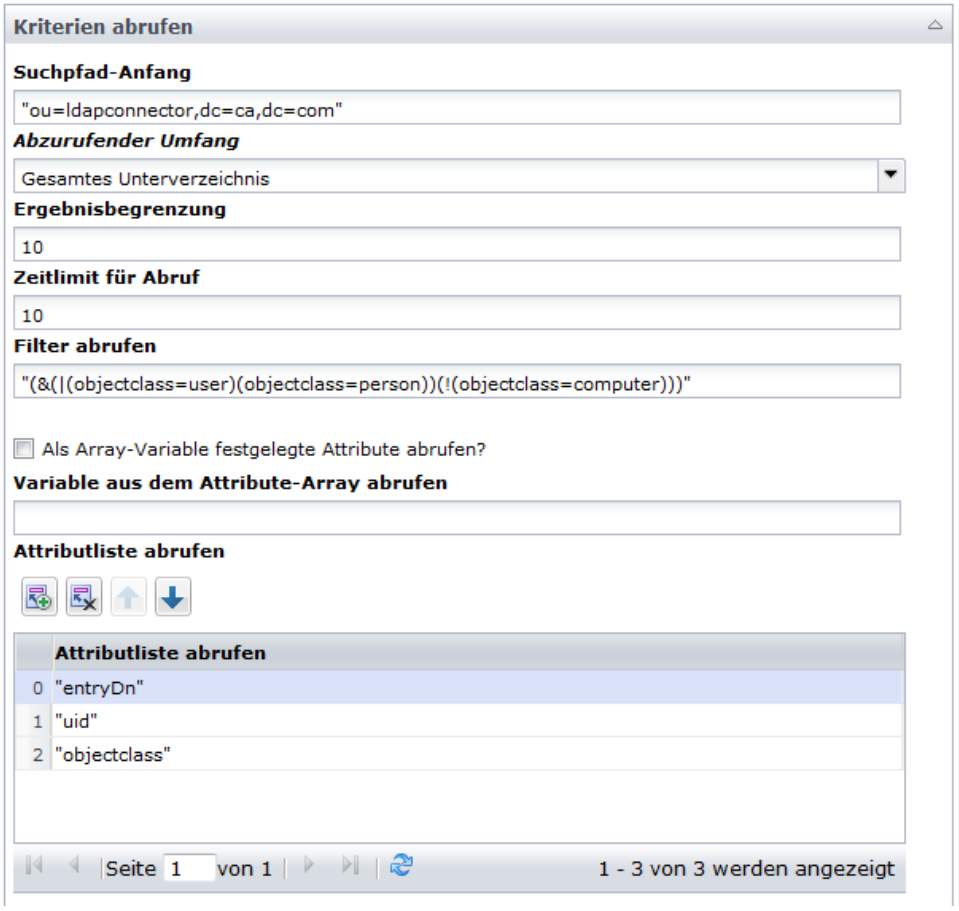
Die meisten LDAP-Server unterscheiden sich in den Anzeigenamen der LDAP-Attribute, aber sie alle müssen die LDAP-Namen dieser Attribute unterstützen. Aus diesem Grund müssen die LDAP-Namen der Attribute statt der Anzeigenamen verwendet werden.

Sie müssen die Namen der abzurufenden Attribute angeben. Andernfalls gibt CA Process Automation in der Variable UserAccounts keine Daten zurück.

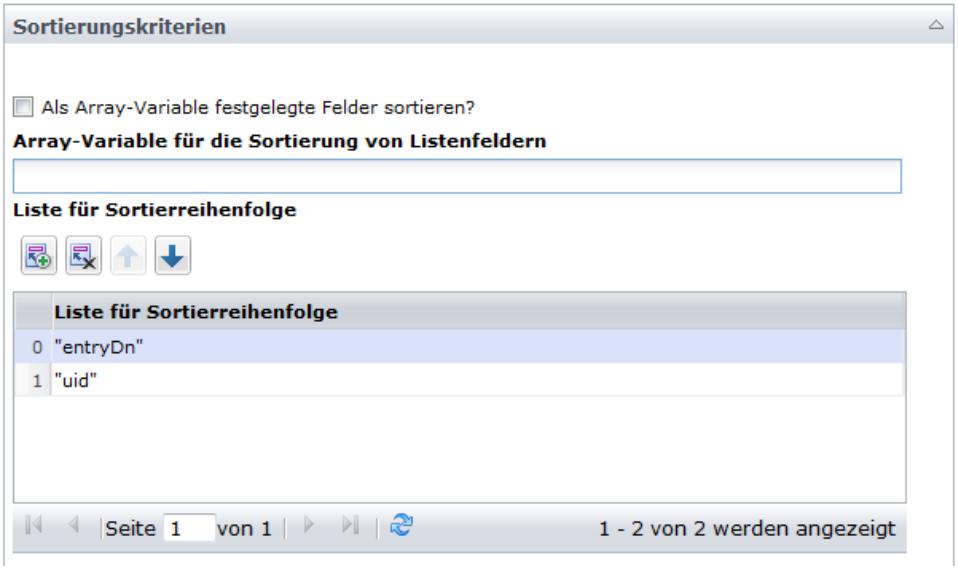
Wenn keine Anwenderkonten unter dem angegebenen Suchweg gefunden wurden, ist die Suchoperation erfolgreich, und die Variable UserAccounts ist leer. Der Suchvorgang schlägt in diesen Fall nicht fehl.

Beispiele

Im diesem Beispiel versuchen wir, alle Anwenderkonten im Pfad "ou=ldapconnector,dc=ca,dc=com" abzurufen. Wir haben insbesondere um die Angabe der Attribute "entrydn", "uid" und "objectclass" dieser Konten gebeten.

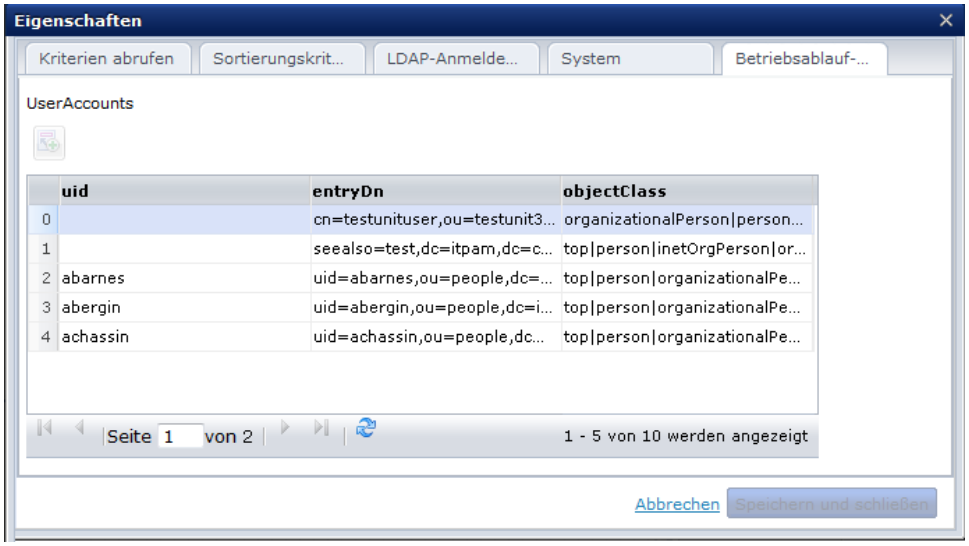


Wir sortieren dann auch die zurückgegebenen Anwenderkonten nach "entryDn" und danach nach "uid".



Nachdem der Operator ausgeführt wurde, wurde seine Verarbeitung erfolgreich abgeschlossen, und die Variable "UserAccounts" wurde folgendermaßen erstellt:

Operation Results	
UserAccounts	[10]
Element Type	
Row0	
uid	
entryDn	
objectClass	
[0]	
Row0	
uid	
entryDn	cn=testunituser,ou=testunit3,dc=itpam,dc=com
objectClass	organizationalPerson|person|top
[1]	
Row0	



Jede ValueMap in der Variable UserAccounts zeigt jetzt die für jedes Anwenderkontoattribut abgerufenen Werte. Der Operator hat nur zehn Anwenderkonten zurückgegeben, weil wir zehn als Suchergebnislimit angegeben haben. Die Werte von mehrwertigen Attribute (in diesem Beispiel "objectclass") werden mit einem "|" zwischen den einzelnen Werten zurückgegeben:

objectClass	organizationalPerson|person|top

Operator-Fehler

Dieser Operator wird in den folgenden Fällen fehlschlagen:

- Der Suchpfad ist auf dem LDAP-Server nicht vorhanden.
- Das Zeitlimit für die Suche wurde überschritten.
- Unfähig, eine Verbindung mit dem LDAP-Server herzustellen.

Je nach LDAP-Server *kann* diese Operation in den folgenden Fällen fehlschlagen, da einige LDAP-Server diese Situation für einen Fehler halten, andere dagegen nicht:

- Der Suchfilter ist ungültig.
- Die zurückgegebenen Attribute sind ungültig.
- Die Sortierattribute sind ungültig.

Operator "Objekt verschieben"

Der Operator "Objekt verschieben" verschiebt ein LDAP-Objekt innerhalb des LDAP-Servers von einem Speicherort zu einem anderen.

Eingabeparameter

Alter Objekt-DN

Gibt den Distinguished Name des Objekts an, das verschoben werden soll.

Neuer Objekt-DN

Gibt das Ziel für den Distinguished Name an, zu dem das Objekt verschoben werden soll.

Ausgabeparameter

LDAPMoveObjectOldDn

LDAPMoveObjectNewDn

remoteLDAPHost

remoteLDAPPort

remoteLDAPUser

remoteLDAPPassword

LDAPBaseDN

LDAPUserPrefix

Beispiel

Operator-Fehler

Dieser Operator wird in den folgenden Fällen fehlschlagen:

- Der alte DN des Objekts existiert nicht im LDAP-Server.
- Der neue DN des Objekts existiert bereits.
- Der LDAP-Server ist so konfiguriert, dass das programmatische Auftreten eines "Verschieben"-Operators verhindert wird.
- Unfähig, eine Verbindung mit dem LDAP-Server herzustellen.

Operator "Anwender aus Gruppe entfernen"

Mit dem Operator "Anwender aus Gruppe entfernen" können Sie einen LDAP-Anwender aus einer LDAP-Gruppe auf dem LDAP-Server entfernen.

Eingabeparameter

Anwender-DN

Gibt den Distinguished Name des Anwenders an, der aus der Gruppe entfernt werden soll.

Group DN

Gibt den Distinguished Name der Gruppe an, aus der der Anwender entfernt werden soll.

Ausgabeparameter

LDAPUserDn

LDAPGroupDn

remoteLDAPHost

remoteLDAPPort

remoteLDAPUser

remoteLDAPPassword

LDAPBaseDN

LDAPUserPrefix

Beispiel

Operator-Fehler

Dieser Operator wird in den folgenden Fällen fehlschlagen:

- Der Anwender ist kein Mitglied der Gruppe.
- Die Gruppe existiert nicht im LDAP-Server.
- Der DN der Gruppe zeigt auf ein LDAP-Objekt, das nicht vom Typ group, groupofnames oder groupofuniquenames ist.
- Unfähig, eine Verbindung mit dem LDAP-Server herzustellen.

Operator "Objektattribute aktualisieren"

Der Operator "Objektattribute aktualisieren" führt die folgenden Aufgaben gleichzeitig aus:

- Hinzufügen neuer Attribute zu einem vorhandenen LDAP-Objekt.
- Ersetzen der Werte von Attributen eines vorhandenen LDAP-Objekts.
- Entfernen der Attribute von einem vorhandenen LDAP-Objekt.

Sie können alle drei Operationen gleichzeitig auf dem gleichen Objekt ausführen, oder Sie können alle Operationen ignorieren und in den Parametern zum Operator nichts eingeben.

Hinweis: Active Directory erlaubt die Änderung eines Anwenderkennworts nur, wenn CA Process Automation über SSL mit dem Active Directory-Server verbunden ist. Wenn CA Process Automation nicht über SSL verbunden ist, schlägt ein Operator "Ersetzen" auf einem Anwenderkennwort fehl.

Weitere Informationen zum Import eines Zertifikats aus Active Directory in CA Process Automation finden Sie unter Hinzufügen eines SSL-Zertifikats zu CA Process Automation (siehe Seite 266). Nachdem das Zertifikat importiert wurde, können Sie die Kategorieeigenschaften der Verzeichnisdienste ändern, um eine SSL-Verbindung mit dem Active Directory-Server herzustellen.

Eingabeparameter

Nachfolgend sind die Eingabeparameter für den Operator "Objektattribute aktualisieren" aufgeführt.

Objektparameter

Eindeutiger Objektname

Gibt den zu ändernden Distinguished Name des LDAP-Objekts an, dessen Attribute Sie ändern wollen.

Seite "Attributparameter hinzufügen"

Verwenden Sie die Seite "Attributparameter hinzufügen", um alle Attribute einzugeben, die Sie für das LDAP-Objekt erstellen möchten.

Angegebene Array-Felder für die hinzuzufügenden LDAP-Attribute verwenden

Aktivieren Sie dieses Feld, um die Attribute- und Attributwerte-Arrays für diese Anfrage zu verwenden.

LDAP-Attribute-Array

Das Array, das den LDAP-Namen der Attribute enthält, die dem Objekt hinzugefügt werden sollen. Dieses Datensatzfeld muss als Array (indizierte Zeichenfolge) definiert werden. Wenn "Angegebene Array-Felder für die hinzuzufügenden LDAP-Attribute verwenden" aktiviert ist, wird dieses Feld verwendet.

LDAP-Attributwerte-Array

Das Array, das die Werte der dem Objekt hinzuzufügenden Attribute enthält. Dieses Datensatzfeld muss als Array (indizierte Zeichenfolge) definiert werden. Wenn "Angegebene Array-Felder für die hinzuzufügenden LDAP-Attribute verwenden" aktiviert ist, wird dieses Feld verwendet.

LDAP Attributes

Die LDAP-Namen der Attribute, die dem Objekt hinzugefügt werden sollen. Wenn "Angegebene Array-Felder für die hinzuzufügenden LDAP-Attribute verwenden" deaktiviert ist, wird dieses Feld verwendet.

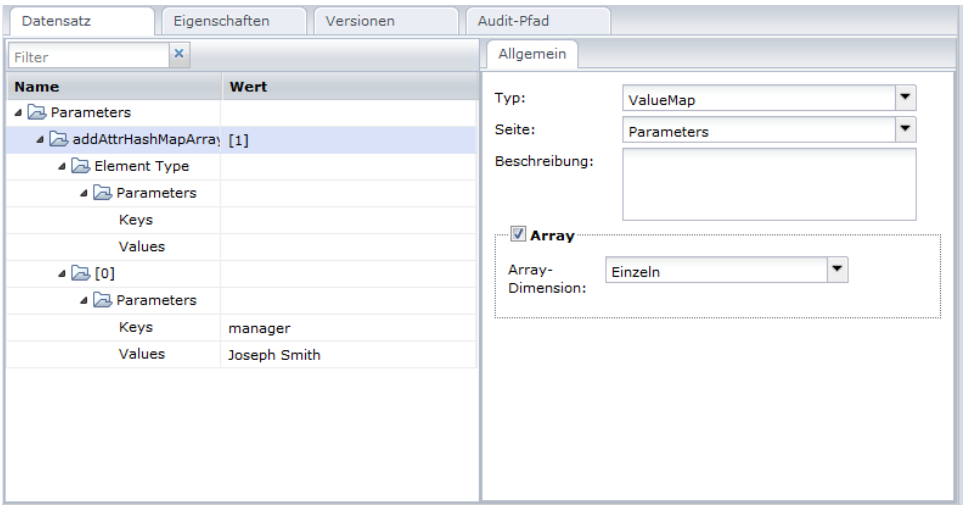
LDAP-Attributwerte

Die Werte der Attribute, die zum Objekt hinzugefügt werden sollen. Wenn "Angegebene Array-Felder für die hinzuzufügenden LDAP-Attribute verwenden" deaktiviert ist, wird dieses Feld verwendet.

LDAP-Attribute für Wertzuordnungen

Dies ist ein Array von Wertzuordnungen, das zusätzliche LDAP-Attribute enthält, die dem Objekt hinzugefügt werden sollen. Der Schlüssel und Wert jeder Wertzuordnung müssen vom Typ "Zeichenfolge" sein; zudem muss der Schlüssel "Key" genannt werden und der Wert muss "Values" genannt werden.

Beispiel



In diesem Beispiel ist das Objekt addAttrHashMapArray eine indizierte ValueMap, deren Schlüsselfelder *Schlüssel* und deren Wertfelder *Werte* genannt werden.

Beachten Sie, dass der Anwender die LDAP-Attribute-Wertzuordnungen allein oder in anderen Feldern auf der Seite als Zusatz zu allen Attributen (und zugeordneten Attributwerten) verwenden kann.

Seite "Parameter zum Ersetzen von Attributen"

Verwenden Sie die Seite "Parameter zum Ersetzen von Attributen", um alle Attribute einzugeben, deren Werte Sie im LDAP-Objekt ersetzen möchten.

Angegebene Array-Felder für die hinzuzufügenden LDAP-Attribute verwenden

Wenn diese Option aktiviert ist, werden für diese Anfrage das Array für Attribute und das Array für Attributwerte verwendet.

LDAP-Attribute-Array

Gibt das Array an, das die LDAP-Namen der Attribute enthält, deren Werte im Objekt ersetzt werden sollen. Dieses Datensatzfeld muss als Array (indizierte Zeichenfolge) definiert werden. Wenn "Angegebene Array-Felder für die hinzuzufügenden LDAP-Attribute verwenden" aktiviert ist, wird dieses Feld verwendet.

LDAP-Attributwerte-Array

Gibt das Array an, das die neuen Werte der Attribute enthält, deren Werte im Objekt ersetzt werden sollen. Dieses Datensatzfeld muss als Array (indizierte Zeichenfolge) definiert werden. Wenn "Angegebene Array-Felder für die hinzuzufügenden LDAP-Attribute verwenden" aktiviert ist, wird dieses Feld verwendet.

LDAP Attributes

Gibt die LDAP-Namen der Attribute an, deren Werte im Objekt ersetzt werden sollen. Wenn "Angegebene Array-Felder für die hinzuzufügenden LDAP-Attribute verwenden" deaktiviert ist, wird dieses Feld verwendet.

LDAP-Attributwerte

Gibt die neuen Werte der Attribute an, deren Werte im Objekt ersetzt werden sollen. Wenn "Angegebene Array-Felder für die hinzuzufügenden LDAP-Attribute verwenden" deaktiviert ist, wird dieses Feld verwendet.

LDAP-Attribute für Wertzuordnungen

Gibt ein Array von Wertzuordnungen an, die die LDAP-Namen und neuen Werte der Attribute enthalten, deren Werte im Objekt ersetzt werden sollen. Für jede Wertzuordnung gilt: Ihr Schlüssel und ihr Wert müssen den Typ "Zeichenfolge" haben, und der Schlüssel muss den Namen "Schlüssel" und der Wert muss den Namen "Werte" haben.

Seite "Parameter zum Entfernen von Attributen"

Die Seite "Parameter zum Entfernen von Attributen" wird zur Eingabe aller Attribute verwendet, die Sie aus dem LDAP-Objekt entfernen möchten.

Angegebenes Array-Feld für die zu entfernenden LDAP-Attribute verwenden

Aktivieren Sie dieses Feld, um Attribute-Arrays für diese Anfrage zu verwenden.

LDAP-Attribute-Array

Gibt das Array an, das die LDAP-Namen der vom Objekt zu entfernenden Attribute enthält. Dieses Datensatzfeld muss als Array (indizierte Zeichenfolge) definiert werden. Wenn "Angegebene Array-Felder für die hinzuzufügenden LDAP-Attribute verwenden" aktiviert ist, wird dieses Feld verwendet.

LDAP Attributes

Gibt die LDAP-Namen der Attribute, die aus dem Objekt entfernt werden sollen. Wenn "Angegebene Array-Felder für die hinzuzufügenden LDAP-Attribute verwenden" deaktiviert ist, wird dieses Feld verwendet.

Ausgabeparameter

LDAPModifyObjAttrsDN

LDAPAddAttributesType

LDAPAddAttributesKeyArray

LDAPAddAttributesValueArray

LDAPAddAttributesKeys

LDAPAddAttributesValues

LDAPAddAttributesMap

LDAPRemoveAttributesType

LDAPRemoveAttributesKeyArray

LDAPRemoveAttributesKeys

LDAPReplaceAttributesType

LDAPReplaceAttributesKeyArray

LDAPReplaceAttributesValueArray

LDAPReplaceAttributesKeys

LDAPReplaceAttributesValues

LDAPReplaceAttributesMap

remoteLDAPHost

remoteLDAPPort

remoteLDAPUser

remoteLDAPPassword

LDAPBaseDN

LDAPUserPrefix

Operator-Fehler

Dieser Operator wird in den folgenden Fällen fehlschlagen:

- Das angegebene LDAP-Objekt existiert nicht.
- Eines der LDAP-Attribute, die Sie hinzufügen, ersetzen oder entfernen möchten, existiert nicht.
- Die Liste von Attributen und Attributwerten der Seite "Hinzufügen" oder "Ändern" stimmen hinsichtlich der Länge nicht überein.
- Der Anwender hat angegeben, dass ein Array von Attributen und ein Array von Attributwerten für die Seiten "Hinzufügen", "Ändern" oder "Entfernen" verwendet werden, aber das in das Arrayfeld eingegebene CA Process Automation-Objekt ist nicht vom Typ Array (indizierte Zeichenfolgen).
- Der LDAP-Server ist so konfiguriert, dass das programmatische Ändern von LDAP-Objekten verhindert wird.
- Unfähig, eine Verbindung mit dem LDAP-Server herzustellen.

Operator "Anwenderstartverzeichnis aktualisieren"

Der Operator "Anwenderstartverzeichnis aktualisieren" richtet eine Freigabe für einen Anwender in einem Active Directory-Server ein. Die Freigabe umfasst ein Stammlaufwerk, Stammverzeichnis und ein Anmeldeskript.

Eingabeparameter

Anwender-DN

Gibt den Distinguished Name des Anwenderobjekts an, für das das Stammverzeichnis, Stammlaufwerk und das Anmeldeskript festgelegt werden soll.

Basisverzeichnis -

Gibt das neue Stammverzeichnis des Anwenders an.

Basislaufwerk

Gibt das neue Stammlaufwerk des Anwenders an.

Anmeldeskript

Gibt das neue Anmeldeskript des Anwenders an.

Ausgabeparameter

LDAPUserDn

LDAPADHomeDirectory

LDAPADHomeDrive

LDAPADLogonScript

remoteLDAPHost

remoteLDAPPort

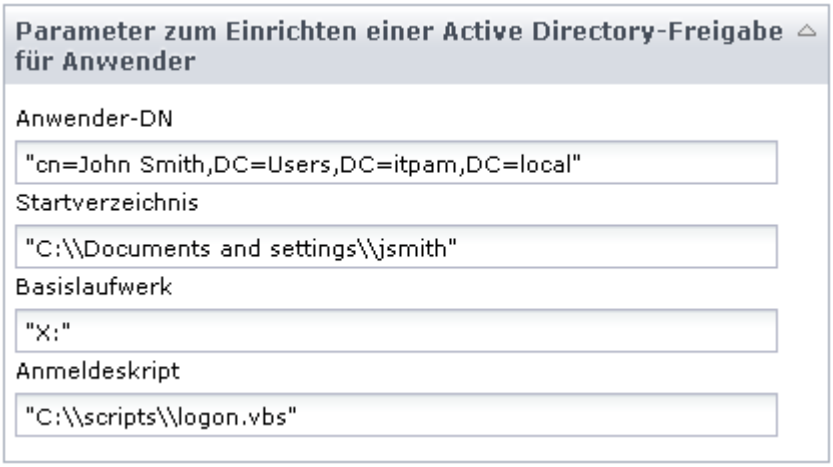
remoteLDAPUser

remoteLDAPPassword

LDAPBaseDN

LDAPUserPrefix

Beispiel



Operator-Fehler

Dieser Operator wird in den folgenden Fällen fehlschlagen:

- Das Anwenderobjekt ist nicht vorhanden.
- Das Active Directory ist so eingerichtet, um programmatische auftretende Modifikationen an LDAP-Objekten zu verhindern.

Hinzufügen eines SSL-Zertifikats zu CA Process Automation

So fügen Sie CA Process Automation ein SSL-Zertifikat hinzu

1. Rufen Sie die Zertifikatsdatei vom Active Directory-Server ab.

   Wenn Sie zum Beispiel eine SSL-Verbindung zwischen CA Process Automation und einem Active Directory-Server herstellen, rufen Sie das Zertifikat ab. Melden Sie sich bei http://*i.p.*/certsrv an, wobei *i.p.* die IP-Adresse des Active Directory-Servers ist, und laden Sie das Zertifikat dann herunter.

2. Kopieren Sie die Zertifikatsdatei auf den Computer, auf dem die CA Process Automation-Verzeichnisdienstoperatoren ausgeführt werden.

3. Importieren Sie das Zertifikat mit dem Befehl "keytool":

   ```
   keytool -import -alias PAM -file certnew.cer -keystore
   "C:\\Programme\\Java\\jdk1,6.0_03\\jre\\lib\\security\\cacerts"
   ```

   Hier steht *certnew.cer* für den Pfad der in Schritt 1 abgerufenen Zertifikatsdatei.

   "C:\\Programme\\Java\\jdk1,6.0_03\\jre\\lib\\security\\cacerts" ist der Pfad der Datei "cacerts" in Java JRE oder JDK.

   - Das Programm "keytool" ist Teil der Java-Installation.
   - "Keytool" fordert zur Eingabe eines Kennworts auf. Das Kennwort lautet standardmäßig "changeit".
   - "Keytool" zeigt zudem die Eingabeaufforderung "Trust this certificate?[no]'" an, um festzustellen, ob das Zertifikat vertrauenswürdig ist. Geben Sie "ja" an.

4. Fügen Sie folgende Zeilen in die CA Process Automation-Datei hinzu:

   ```
   PAM\server\c2o\bin\c2osvcw.conf
   ```

   (oder im Fall eines Upgrade): I

   ```
   PAM_DIR%\server\c2o\bin\c2osvcw.conf:
   ```

   ```
   wrapper.java.additional.11=-Djavax.net.ssl.trustStore="C:\Programme\Java\jdk1
   .6.0_03\jre\lib\security\cacerts"
   wrapper.java.additional.12=-Djavax.net.ssl.trustStorePassword="changeit"
   ```

   Möglicherweise müssen Sie andere Nummern angeben. Fangen Sie mit der weiter verfügbaren Nummer an. Wenn wrapper.java.additional.11 bereits definiert ist, verwenden Sie 12 und 13.

   Der Programmordner ist bei Ihrer JDK-Installation auch ein anderer.

   Das Standardkennwort lautet "changeit".

5. Starten Sie den CA Process Automation-Kontaktpunkt neu, der die Verzeichnisdienste-Operatoren enthält.

Einrichten eines Active Directory-Server

Um eine SSL-Verbindung zwischen den CA Process Automation-Verzeichnisdienstoperatoren und einem Active Directory-Server herzustellen, stellen Sie sicher, dass der Active Directory-Server folgendermaßen konfiguriert ist:

1. Die Zertifikatdienste sind auf dem Active Directory-Server installiert (wenden Sie sich für diese Aufgabe an Ihren Active-Directory-Administrator).
2. Für den Domänen-Controller ist das automatisch Anfordern von Zertifikaten konfiguriert (wenden Sie sich für diese Aufgabe an Ihren Active-Directory-Administrator).

Hinweis: Wenn ein neues Anwenderkonto erstellt oder das Kennwort eines vorhandenen Anwenderkontos in Active Directory geändert wird, lässt Active Directory nur dann zu, dass Sie ein Anwenderkennwort erstellen oder ändern, wenn CA Process Automation über eine SSL-Verbindung mit dem Active Directory-Server verfügt.

Kapitel 8: E-Mail

Die E-Mail-Operatoren können Aufgaben automatisieren, die auf Meldungen und Ordnern in einem E-Mail-Server ausgeführt werden. E-Mail-Operatoren kommunizieren mit Ihrem Mail-Server bei der entfernten Verwendung von einem der folgenden Protokolle:

Post Office Protocol - Version 3 (POP3)

POP3, das für den Zugriff auf ein Postfach durch Einzelanwender geeignet ist, ermöglicht es Ihnen, E-Mail-Nachrichten auf Ihren lokalen Computer herunterzuladen. Standardmäßig hören POP3-Server TCP-Port 110 ab.

POP-SSL

Standardmäßig lauschen POP-SSL-Server auf TCP-Port 995.

Internet Message Access Protocol (IMAP)

IMAP, das für den Zugriff auf ein Postfach durch mehrere Anwender geeignet ist, ermöglicht gleichzeitigen Zugriff durch mehrere Clients. Standardmäßig lauschen IMAP-Server auf TCP-Port 143.

IMAP-SSL

Standardmäßig lauschen IMAP-Server auf TCP-Port 993.

Die E-Mail-Operatoren können nur mit Ihrem Mailserver kommunizieren, wenn entweder IMAP oder POP3 unterstützt wird. Die E-Mail-Operatoren führen Aktionen, beispielsweise das Abrufen von E-Mail-Anzahlen. Beide Protokolle unterstützen diesen Aktionstyp. Einige Aktionen, wie z. B. "Ordner löschen", werden nur vom IMAP-Protokoll unterstützt.

Allgemeine Parameter für E-Mail-Operatoren

Die folgenden Eigenschaften gelten für verschiedene E-Mail-Operatoren:

- Filterkriterien für Meldungen
- Anmeldeparameter für den Mail-Server

Filterkriterien für Meldungen

Meldungs-ID

Gibt die eindeutige ID der zu filternden E-Mail an. Sie können auch eine untergeordnete Zeichenfolge der Meldungs-ID übergeben. Wenn die IDs der gewünschten E-Mails nicht bekannt sind, können Sie die Meldungs-IDs von Meldungen mithilfe des Operators "E-Mail-Liste abrufen" (siehe Seite 284) abrufen. Die Aktion wird für alle E-Mails ausgeführt, die mit der untergeordneten Zeichenfolge des Betreffs übereinstimmen.

Hinweis: Dieser Parameter ist nicht für alle E-Mail-Operatoren verfügbar.

Meldungsanzahl

Gibt die Meldungsnummer der zu filternden E-Mail an. Dieser Parameter ist *keine* statische Anzahl für eine Meldung. Wenn E-Mails gelöscht und in andere Ordner verschoben werden, ändert sich die Meldungsnummer. Wenn parallele Aktionen im gleichen Postfachordner stattfinden, empfehlen wir Ihnen, "Meldungs-ID" (statt "Meldungsanzahl") zu verwenden, um Meldungen anzugeben.

Hinweis: Dieser Parameter ist nicht für alle E-Mail-Operatoren verfügbar.

Meldungsbetreff

Gibt den Betreff der E-Mail an, der übereinstimmen soll. Dieser Parameter kann eine untergeordnete Zeichenfolge oder ein regulärer Ausdruck sein. Die Aktion wird für alle E-Mails ausgeführt, die mit der untergeordneten Zeichenfolge des Betreffs übereinstimmen.

Meldungssender

Gibt den E-Mail-Absender an, der übereinstimmen soll. Dieser Parameter kann eine untergeordnete Zeichenfolge oder ein regulärer Ausdruck sein. Die Aktion wird für alle E-Mails ausgeführt, die mit der untergeordneten Zeichenfolge des Betreffs übereinstimmen.

Werte für Meldungsbetreff und Absender der Meldung sind reguläre Ausdrücke

Aktivieren Sie dieses Kontrollkästchen, um anzugeben, dass die Werte in "Meldungsbetreff" und "Meldungssender" beim Filtern von E-Mails als reguläre Ausdrücke (anstatt als einfache Zeichenfolge) behandelt werden.

Wichtig! Um E-Mails für die E-Mail-Operatoren mithilfe von regulären Ausdrücken zu analysieren, müssen *alle* E-Mails aus dem Postfach abgerufen und auf dem Client analysiert werden. Das heißt, auf dem Kontaktpunkt, auf dem der Operator ausgeführt wird. Wir empfehlen Ihnen, dieses Feld nicht auszuwählen und reguläre Ausdrücke nur zu verwenden, wenn es erforderlich ist.

Text

Wird als untergeordnete Zeichenfolge des E-Mail-Texts angegeben, der übereinstimmen soll.

Sendezeitpunkt der ersten Meldung

Übereinstimmung von E-Mails, die nach der angegebenen Uhrzeit gesendet wurden. Dieser Parameter identifiziert die früheste Uhrzeit, zu der die abzugleichende E-Mail gesendet wurde, angegeben in einer CA Process Automation-Variablen.

Sendezeitpunkt der letzten Meldung

Übereinstimmung mit Meldungen, die vor der angegebenen Zeit gesendet wurden. Dieser Parameter identifiziert die späteste Uhrzeit, zu der die übereinzustimmende Meldung gesendet und in einer CA Process Automation-Variablen angegeben wurde.

Hinweis: Die Felder "Sendezeitpunkt der ersten Meldung" und "Sendezeitpunkt der letzten Meldung" sind CA Process Automation-Datentypvariablen. Systemfunktionen wie now() (siehe Seite 599) oder today() (siehe Seite 606) generieren Datentypvariablen. Die Systemfunktion parseDate (siehe Seite 600) (stringDate, simpleDateFormat) erstellt das Datum richtig. "stringDate" ist ein Datum im Zeichenfolgenformat, und "simpleDateFormat" ist das Format, das verwendet werden soll, um das Datum zu analysieren. Diese Funktion kann verwendet werden, um eine Zeichenfolge in eine CA Process Automation-Datentypvariable zu analysieren. Zum Beispiel:

```
parseDate("2010/07/28 13:00:01", "yyyy/MM/dd HH:mm:ss")
```

IMAP-Meldungs-Flag

IMAP verwendet Meldungs-Flags, um den Zustand einer E-Mail zu überwachen. Diese Flags werden auf dem Server gespeichert. Verschiedene Clients, die an unterschiedlichen Zeiten auf das gleiche Postfach zugreifen, können die Änderungen erkennen, die andere Clients verursachen.

Die folgenden Flags sind gültig und können programmatisch festgelegt werden, indem die Werte auf die Namen der Flags festgelegt werden. Der Name der Flags unterscheidet Groß- und Kleinschreibung und wird in Großbuchstaben übergeben.

Wählen Sie das Flag aus, das Sie für Ihre Meldung festlegen möchten. Diese Flags funktionieren zusammen mit dem Kontrollkästchen "IMAP-Meldungs-Flag ist auf 'wahr' gesetzt".

BEANTWORTET

Wenn Sie das Kontrollkästchen "IMAP-Meldungs-Flag ist auf 'wahr' gesetzt" auswählen, werden nur beantwortete E-Mails zurückgegeben (die E-Mails, zu denen die Antworten gesendet wurden).

GELÖSCHT

Wenn Sie das Kontrollkästchen IMAP-Meldungs-Flag ist auf 'wahr' gesetzt aktivieren, werden alle E-Mails zurückgegeben, die nicht als vom Server als gelöscht markiert sind.

ENTWURF

Wenn Sie das Kontrollkästchen "IMAP-Meldungs-Flag ist auf 'wahr' gesetzt" auswählen, werden alle E-Mails zurückgegeben, die sich derzeit nicht in einem Entwurfsstatus befinden.

MARKIERT

Wenn Sie das Kontrollkästchen "IMAP-Meldungs-Flag ist auf 'wahr' gesetzt" auswählen, werden alle nicht markierten E-Mails zurückgegeben.

LETZTE

Wenn Sie das Kontrollkästchen "IMAP-Meldungs-Flag ist auf 'wahr' gesetzt" auswählen, werden alle ungeöffneten (nicht geklickten) E-Mails zurückgegeben.

GESEHEN

Wenn Sie das Kontrollkästchen IMAP-Meldungs-Flag ist auf 'wahr' gesetzt aktivieren, werden alle als gelesen markierten E-Mails zurückgegeben. Wenn Sie das Kontrollkästchen *IMAP-Meldungs-Flag ist auf 'wahr' gesetzt* nicht aktivieren, werden alle als ungelesen markierten E-Mails zurückgegeben.

IMAP-Meldungs-Flag ist auf 'wahr' gesetzt

Der IMAP-Flag-Name ist in der Liste "IMAP-Meldungs-Flag" festgelegt. Der zugehörige boolesche Wert kann aus diesem Kontrollkästchen ausgewählt werden. Wenn Sie zum Beispiel die Option "Gesehen" als "wahr" festlegen möchten, dann wählen Sie "Gesehen" aus dem Drop-down-Feld aus, und wählen Sie dann dieses Kontrollkästchen aus.

Anmeldeparameter für den Mail-Server

Protokoll für Verbindung

Wählen Sie das E-Mail-Protokoll aus, das verwendet werden soll, um eine Verbindung mit dem Server herzustellen:

- IMAP
- IMAP-SSL
- POP3
- POP3-SSL

Mailserver-Host

Gibt den Hostnamen oder die IP-Adresse des Mailservers an.

Mailserver-Port

Gibt den Port des Mailservers an.

Anwendername

Gibt den Anwendernamen des Anwenders an für den Zugriff auf den Mailserver an.

Kennwort

Gibt das Kennwort des Anwenders für den Zugriff auf den Mailserver an.

Operator "Ordner erstellen"

Der Operator "Ordner erstellen" erstellt einen Ordner auf dem Mail-Server. Ordner werden rekursiv mittels des IMAP-Protokolls erstellt.

Wenn der Ordner existiert, wird eine Ausnahme mit dem Text "Ordner existiert bereits" ausgegeben.

Eingabeparameter

Postfachordner-Name

Name des Ordners, der im Mailserver erstellt werden soll.

Anmeldeparameter für den Mail-Server **(siehe Seite 272)**

Ausgabeparameter

FolderCreated

Gibt "wahr" zurück, wenn der Ordner erfolgreich erstellt wurde, und gibt andernfalls "falsch" zurück.

FolderName

Protokoll

ServerHost

ServerPort

UserName

Kennwort

Operator "E-Mail löschen"

Der Operator "E-Mail löschen" löscht Meldungen aus dem Postfach und gibt die Anzahl der gelöschten Meldungen zurück. Dieser Operator verwendet das IMAP-Protokoll.

Hinweis: Wenn alle Felder leer gelassen werden, löscht dieser Operator alle Meldungen aus dem Postfach (angegebener Postfachordnername).

Eingabeparameter

Postfachordner-Name

Gibt den Namen des Ordners an, der Meldungen enthält, die im Mailserver gelöscht werden sollen.

Filterkriterien für Meldungen **(siehe Seite 270)**

Anmeldeparameter für den Mail-Server **(siehe Seite 272)**

Ausgabeparameter

DeletedCount

Gibt die Anzahl der gelöschten Nachrichten zurück.

FolderName

MessageID

MessageNumber

Betreff

Von:

IsRegExp

Text

SentFromDate

SentToDate

FlagField

FlagValue

Protokoll

ServerHost

ServerPort

UserName

Kennwort

Operator "Ordner löschen"

Der Operator "Ordner löschen" löscht einen Ordner auf dem Server mithilfe des IMAP-Protokolls. Der Ordner wird gelöscht, auch wenn er Unterordner enthält.

Eingabeparameter

Postfachordner-Name

Gibt den Namen des Ordners an, der auf dem Mailserver gelöscht werden soll.

Anmeldeparameter für den Mail-Server **(siehe Seite 272)**

Ausgabeparameter

FolderCreated

Gibt "Wahr" zurück, wenn der Ordner erfolgreich gelöscht wurde, und andernfalls "Falsch".

FolderName

Protokoll

ServerHost

ServerPort

UserName

Kennwort

Beispiel

Wichtig! Im folgenden Szenario wird ein Hmail-Mailserver verwendet. Dieses Szenario funktioniert nicht, wenn ein Domino-Server verwendet wird.

1. Erstellen Sie eine Hierarchie von Ordnern, beispielsweise test1\test2\test3\test4 (parallel zu "Posteingang").
2. Erstellen Sie eine weitere Hierarchie: test2\test5\test6 (parallel zu "Posteingang").
3. Löschen Sie den Ordner "test2" mit dem Operator "Ordner löschen".

Erwartetes Ergebnis

Die Hierarchie "test2\test5\test6" wird gelöscht.

Tatsächliches Ergebnis

Der Prozess wird erfolgreich ausgeführt und gibt "falsch" zurück. Idealerweise wird "Wahr" zurückgegeben. Der Ordner existiert noch auf dem Mail-Server.

Operator "E-Mail-Inhalt abrufen"

Der Operator "E-Mail-Inhalt abrufen" ruft den E-Mail-Text und die Anhänge ab. Wenn das Feld "Meldungsanzahl" weder leer noch gleich 0 ist, wird eine einzelne Nachricht vom Operator abgerufen (entsprechend der Meldungsanzahl) und deren Details zurückgegeben. Andernfalls gibt der Operator den Inhalt aller E-Mails innerhalb des Ordners zurück.

Hinweis: Enthält keines der Felder einen Wert, ruft dieser Operator die Inhalte aller E-Mails aus dem angegebenen Postfachordner ab.

Eingabeparameter

Postfachordner-Name

Gibt den Namen des Ordners an, der die zu verarbeitenden E-Mails enthält.

Abgerufene Meldungen als gesehen festlegen

Wenn diese Option aktiviert ist, werden abgerufene Nachrichten als gesehen festgelegt.

Startindex des E-Mail-Inhalts, der aus der Datensatzvariablen abgerufen werden soll

Wenn Sie in dieses Feld einen Index eingeben, wird der ab diesem Index beginnende Inhalt in der Datensatzvariablen angezeigt. Leer lassen, um am Anfang zu starten.

Länge des E-Mail-Inhalts, der aus der Datensatzvariablen abgerufen werden soll

Wenn Sie in dieses Feld einen Index eingeben, wird der Inhalt bis zu diesem Index in der Datensatzvariablen angezeigt. Geben Sie nichts ein, um den größtmöglichen E-Mail-Inhalt abzurufen.

E-Mail-Anhänge verarbeiten

Wenn ausgewählt, werden auch Anhänge verarbeitet. Die Option ist standardmäßig deaktiviert.

Anhangs-Betriebsablauf

Gibt eine der folgenden Operationen an, die für den Anhang ausgeführt werden können. Diese Option wird aktiviert, wenn der Anwender das Kontrollkästchen "E-Mail-Anhänge verarbeiten" aktiviert.

Anhang in eine Datei speichern

Speichert den Anhang im Zielordner.

Anhangsinhalt in Datensatzvariable abrufen

Schreibt die Anhangsinhalte in eine Datensatzvariable.

Die Eigenschaft zum Abrufen von E-Mail-Inhalten wird als konfigurierbare Eigenschaft dargestellt. Die Standardgröße ist 64 K. Fügen Sie in OasisConfig.properties "mail.maximum.content.size=64" hinzu, um die Größe zu konfigurieren. Die Einheit für die Größe ist K.

Beides

Führt sowohl die Operation "Speichern" als auch die Operation "Inhalte in Datensatzvariable schreiben" aus.

Zielordnername

Gibt den Zielordner an, in dem der Anhang gespeichert wird.

Eindeutige Dateinamen zum Speichern von Anhängen generieren

Wenn diese Option aktiviert ist, wird die Option bereitgestellt, um beim Speichern von Anhängen eindeutige Dateinamen zu generieren.

Startindex des Anhangsinhalts, der aus der Datensatzvariablen abgerufen werden soll

Wenn Sie in dieses Feld einen Index eingeben, wird der ab diesem Index beginnende Anhangsinhalt in der Datensatzvariablen angezeigt. Leer lassen, um am Anfang zu starten.

Länge des Anhangsinhalts, der aus der Datensatzvariablen abgerufen werden soll

Wenn Sie in dieses Feld einen Index eingeben, wird der Anhangsinhalt bis zu diesem Index in der Datensatzvariablen angezeigt. Geben Sie nichts ein, um den größtmöglichen E-Mail-Inhalt abzurufen.

Filterkriterien für Meldungen (siehe Seite 270)

Anmeldeparameter für den Mail-Server (siehe Seite 272)

Ausgabeparameter

MessageContent

(ValueMap) Gibt die Nachrichten- und Anhangsinhalte zurück.

Enthält:

ResultRow

AttachmentContents

MailContents

AttachmentFiles

FolderName

MarkMessagesAsSeen

MessageContentStartIndex

MessageContentLength

IsProcessAttachement

ProcessAttachmentType

DestinationFolderName

GenerateUniqueName

MessageAttachmentStartIndex

MessageAttachmentLength

MessageID

MessageNumber

Betreff

Von:

IsRegExp

Text

SentFromDate

SentToDate

FlagField

FlagValue

Protokoll

ServerHost

ServerPort

UserName

Kennwort

Operator "E-Mail-Anzahl abrufen"

Der Operator "E-Mail-Anzahl abrufen" gibt die Anzahl an E-Mails im Ordner zurück.

Sie können entweder über das POP3-Protokoll oder das IMAP-Protokoll eine Verbindung mit dem Mail-Server herstellen. Je nach Protokoll muss der Anwender die entsprechende Portnummer angeben.

- Der Standardport für POP3 ist 110.
- Der Standardport für IMAP ist 143.
- Der Standardport für POP-SSL ist 995.
- Der Standardport für IMAP-SSL ist 993.

Eingabeparameter

Postfachordner-Name

Gibt den Namen des Ordners an, der die zu verarbeitenden E-Mails enthält.

Filterkriterien für Meldungen (siehe Seite 270)

Anmeldeparameter für den Mail-Server (siehe Seite 272)

Ausgabeparameter

MessageCount

Gibt die Anzahl von E-Mails im Ordner zurück.

FolderName

MessageID

MessageNumber

Betreff

Von:

IsRegExp

Text

SentFromDate

SentToDate

FlagField

FlagValue

Protokoll

ServerHost

ServerPort

UserName

Kennwort

Operator "E-Mail-Umschlag abrufen"

Der Operator "E-Mail-Umschlag abrufen" ruft die E-Mail-Umschläge vom angegebenen Filterkriterium ab. Wenn die Meldungsnummer nicht leer oder 0 ist (in den Filterkriterien für Meldungen (siehe Seite 270) angegeben), dann ruft dieser Operator eine einzelne E-Mail (basierend auf der Meldungsnummer) ab und gibt die Details zurück. Andernfalls gibt der Operator die Umschläge aller im Ordner enthaltenen E-Mails zurück.

Hinweis: Enthält keines der Felder einen Wert, ruft dieser Operator die Inhalte aller E-Mails aus dem angegebenen Postfachordner ab.

Eingabeparameter

Postfachordner-Name

Gibt den Namen des Ordners an, der die zu verarbeitenden E-Mails enthält.

Abgerufene Meldungen als gesehen festlegen

Wenn diese Option aktiviert ist, werden abgerufene E-Mails als gesehen festgelegt.

Filterkriterien für Meldungen (siehe Seite 270)

Anmeldeparameter für den Mail-Server (siehe Seite 272)

Ausgabeparameter

MessageEnvelope

(ValueMap) Gibt den Umschlag von im Ordner enthaltenen Nachrichten zurück.

Enthält:

ResultRow

SentDate

Betreff

An

Bcc

CC

Von:

FolderName

MarkMessagesAsSeen

MessageID

MessageNumber

Betreff

Von:

IsRegExp

Text

SentFromDate

SentToDate

FlagField

FlagValue

Protokoll

ServerHost

ServerPort

UserName

Kennwort

Operator "E-Mail-Liste abrufen"

Der Operator "E-Mail-Liste abrufen" ruft eine Liste von E-Mails ab, die bestimmten Filterkriterien entsprechen. Sie können die im Abschnitt "Filterkriterien für Meldungen (siehe Seite 270)" beschriebenen Felder so konfigurieren, dass nur die E-Mails herausgefiltert werden, deren Umschläge abgerufen werden sollen. Dieser Operator kann die Protokolle IMAP und POP3 verwenden. Verwenden Sie den Operator "E-Mail-Liste abrufen" insbesondere zum Abrufen von grundlegenden Informationen zu E-Mails. Zu diesen Informationen gehören "Meldungs-ID" und "Meldungsanzahl", die in anderen Operatoren verwendet werden können.

Hinweis: Der Operator "E-Mail-Liste abrufen" kann in einer Iteration die Umschläge von maximal 512 E-Mails abrufen.

Eingabeparameter

Postfachordner-Name

Gibt den Namen des Zielpostfachordners an, der die zu verarbeitenden E-Mails enthält. Dieses Feld darf nicht leer bleiben.

Filterkriterien für Meldungen (siehe Seite 270)

Anmeldeparameter für den Mail-Server (siehe Seite 272)

Ausgabeparameter

MessageList

Ein Array von "ValueMaps". Jeder Index des Arrays ist ein CA Process Automation-ValueMap-Datentyp. Die ValueMap enthält die folgenden Felder mit folgenden Informationen einer einzelnen Nachricht:

MessageID

UniqueID der Meldung.

MessageNumber

Die Meldungsnummer einer E-Mail kann sich ändern, wenn E-Mails zwischen Ordnern verschoben, gelöscht oder dauerhaft beseitigt werden. Wir empfehlen Ihnen, "MessageID" zu verwenden, um E-Mails eindeutig anzugeben.

Betreff

Betreff der E-Mail.

SenderAddress

Adresse des Absenders der E-Mail.

SentDate

Eine CA Process Automation-Datumstypvariable mit dem Datum, an dem die E-Mail gesendet wurde.

ReceivedDate

Eine CA Process Automation-Datumstypvariable mit dem Datum, an dem der Server die E-Mail erhalten hat. Dieser Wert wird nur eingetragen, wenn mithilfe des Protokolls IMAP eine Verbindung mit dem Server hergestellt wird.

NumOfAttachments

Eine Variable vom Typ "Ganzzahl", die die Anzahl der Anhänge einer E-Mail angibt.

FolderName

Betreff

Von:

IsRegExp

Text

SentFromDate

SentToDate

FlagField

FlagValue

Protokoll

ServerHost

ServerPort

UserName

Kennwort

Operator "E-Mail verschieben"

Der Operator "E-Mail verschieben" verschiebt die E-Mails von einem Ordner zu einem anderen.

Eingabeparameter

Postfach-Quellordnername

Name des Quellordners, der die zu verschiebenden E-Mails enthält.

Postfach-Zielordnername

Gibt den Namen des Zielordners an, in den die E-Mails kopiert werden.

Hinweise:

- Wenn der Quellordner nicht existiert, gibt CA Process Automation eine Ausnahme "Quellordner existiert nicht" aus.
- Wenn der Zielordner nicht existiert, erstellt CA Process Automation den Zielordner und verschiebt dann die E-Mails aus dem Quellordner in den Zielordner.

Filterkriterien für Meldungen **(siehe Seite 270)**

Anmeldeparameter für den Mail-Server **(siehe Seite 272)**

Ausgabeparameter

MovedCount

SourceFolderName

DestinationFolderName

MessageID

MessageNumber

Betreff

Von:

IsRegExp

Text

SentFromDate

SentToDate

FlagField

FlagValue

Protokoll

ServerHost

ServerPort

UserName

Kennwort

Operator "Ordner bereinigen"

Der Operator "Ordner bereinigen" beseitigt (entfernt dauerhaft) Ordner, die als GELÖSCHT markiert sind, und gibt die Anzahl der beseitigten E-Mails zurück. Dieser Operator verwendet das IMAP-Protokoll.

Eingabeparameter

Postfachordner-Name

Gibt den Namen des Ordners auf dem Mailserver an, der dauerhaft zu löschende E-Mails enthält.

Anmeldeparameter für den Mail-Server **(siehe Seite 272)**

Ausgabeparameter

ExpungedCount

Gibt die Anzahl der gelöschten E-Mails zurück.

FolderName

Protokoll

ServerHost

ServerPort

UserName

Kennwort

Operator "Ordner umbenennen"

Der Operator "Ordner umbenennen" benennt den Ordner im Mailserver um. Der Operator verwendet das IMAP-Protokoll.

Eingabeparameter

Aktueller Postfachordner-Name

Gibt den alten Namen des umzubenennenden Ordners an.

Neuer Postfachordner-Name

Gibt den neuen Namen für den Ordner an.

Anmeldeparameter für den Mail-Server **(siehe Seite 272)**

Ausgabeparameter

FolderRenamed

Gibt "wahr" zurück, wenn der Ordner erfolgreich umbenannt wurde, und andernfalls "falsch".

OldFolderName

NewFolderName

Protokoll

ServerHost

ServerPort

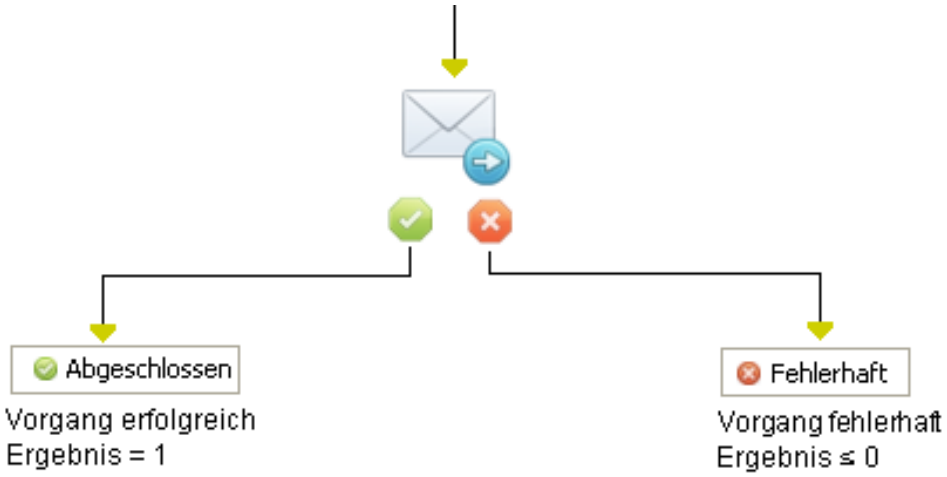
UserName

Kennwort

Operator "E-Mail senden"

Verwenden Sie den Operator "E-Mail senden", um E-Mail-Benachrichtigungen an angegebene Empfänger zu senden.

Die E-Mail-Implementierung von CA Process Automation unterstützt das Senden von E-Mail über einen SMTP-Server. Sie können eine zulässige vom SMTP-Server unterstützte E-Mail-Adresse angeben. Zu diesen Adressen können Aliase, eine Verteilerliste, eine Faxadresse oder ein digitaler Pager gehören. Der SMTP-Server verarbeitet, anstatt CA Process Automation, die tatsächliche Lieferung einer Meldung.



Der Operator "E-Mail senden" kann angehängte Dateien enthalten. Sie können Anhänge angeben, um dynamisch aktualisierte Dateien statt statischer Informationen zu senden, die zu dem Zeitpunkt angegeben wurden, als der Operator "E-Mail senden" einem Prozess hinzugefügt wurde. Diese Funktion ist nützlich zum Senden von Anhängen wie Protokoll- und Ausnahmedateien.

Der Operator "E-Mail senden" unterstützt auch HTTP-/HTTPS-URLs als ein Pfad von Anhängen. Der Operator "E-Mail senden" schlägt fehl, wenn ein Pfadname zur Laufzeit nicht zulässig ist.

Tipp: Geben Sie die Speicherorte für Anhänge bezogen auf den Kontaktpunkt an, der die Operatoren "E-Mail" ausführt.

Eingabeparameter

Anwendername

Gibt einen zulässigen Anwendernamen oder ein zulässiges Profil zum Nachrichtenversand auf dem SMTP-Server an. Beispiel: Process.Email.*Anwendername.*

Kennwort

Gibt das Kennwort für den Anwendernamen an. Beispiel: Process.Email.*Kennwort*.

Von:

Die Adresse, die im Absenderfeld der ausgehenden E-Mail angezeigt wird.

An

Gibt die Adressen von E-Mail-Empfängern an. Trennen Sie mehrere E-Mail-Adressen durch Kommas oder Semikolons voneinander. Beispiel: support@PAM.com,sales@PAM.com.

CC

Gibt die Adressen von Empfängern an, die Kopien dieses E-Mail-Alarms erhalten. Trennen Sie mehrere E-Mail-Adressen durch Kommas oder Semikolons voneinander. Beispiel: system_administrator@PAM.com; support@PAM.com.

Betreff

Gibt eine kurze Beschreibung an, die im Betreff der E-Mail angezeigt werden soll. Beispiel: CA Process Automation-Alarm.

Meldung

Gibt die von der E-Mail gelieferte Meldung an. Beispiel:

```
Hinweis
Backup-Probleme auf einem Kontaktpunkt: + Process.Kontaktpunktname.
```

In HTML-Format senden

Wenn diese Option ausgewählt ist, verwendet CA Process Automation HTML-Text im Feld "Meldung". Wenn diese Option nicht ausgewählt ist, verwendet CA Process Automation Nur-Text-Format.

Verwenden Sie zum Beispiel folgenden Text, um "Hinweis" in roter Schriftart anzuzeigen:

```
<font size="5" color="red">Notice</font>
```

Anhang

Gibt die vollständigen Pfade für Dateien an, die an die E-Mail angehängt werden sollen. Trennen Sie mehrere Pfadnamen durch Kommas oder Semikolons voneinander. Sie können auch HTTP/HTTPS-URLs für lokale und externe Speicherorte angeben.

Dieser Parameter muss Speicherorte angeben, die während der Laufzeit und auf dem Kontaktpunkt, auf dem die E-Mail-Operatoren ausgeführt werden, gültig sind. Beispiel: C:\\CA Process Automation\\Data\\Log\\Global.log.

Empfang

Aktivieren Sie dieses Kontrollkästchen, um eine Zustellbestätigung für die Nachricht anzufordern. Der Empfang ist normalerweise ein Service, den der E-Mail-Client des Empfängers angibt. CA Process Automation kann den Empfang nicht garantieren.

Codierung

Gibt das Kodierungsschema des Textes an, den Leser erhält (UTF-8, UTF-16, US-ASCII, Windows-1250, Windows-1252, Shift_JIS).

Ausgabeparameter

Anwender

password

from

to

cc

subject

text

isContentHtml

Anhang

recpt

Kapitel 9: Dateimanagement

Die Dateimanagement-Operatoren überwachen Verzeichnisse, Dateien und deren Inhalte. Diese Operatoren können entweder lokal oder auf einem Remote-System ausgeführt werden. Die Dateimanagement-Operatoren unterstützen auch Vorgänge auf einem Proxyknoten. Abhängig vom Ergebnis des Vorgangs, wird der Prozess erfolgreich ausgeführt oder schlägt fehl.

Verwenden Sie die Dateimanagement-Operatoren, um lokale Dateien zu erstellen, zu löschen, umzubenennen, zu komprimieren und zu dekomprimieren. Sie können auch die Dateimanagement-Operatoren verwenden, um Dateien auf dem Kontaktpunkt anzuzeigen, auf dem Operatoren ausgeführt werden. Alle Dateimanagement-Operatoren werden unter demselben Anwendernamen ausgeführt, der den Kontaktpunkt ausführt, z. B. Administrator auf einem Windows-Kontaktpunkt oder Root auf einem UNIX-Kontaktpunkt.

Die Dateimanagement-Operatoren können auf Proxy-Agenten ausgeführt werden. Normal- und Proxy-Service verfügen über die gleichen Eigenschaften, aber ein unterschiedliches Verhalten. Wenn beispielsweise die Option zum Protokollieren von Ausgaben für die Ausführung des Prozess-Proxy-Service aktiviert worden ist, wird die Protokolldatei in dem Dateisystem erstellt, in dem der Agent ausgeführt wird. Die Protokolldatei wird nicht auf dem Computer erstellt, auf dem sich der Proxy-Agent befindet.

Wichtig! Die folgenden Bedingungen gelten für alle Operatoren in dieser Kategorie, wenn sie auf einem Remote-Windows-Host über einen Proxy-Kontaktpunkt ausgeführt werden:

- Verwenden Sie Pfade im UNIX-Stil für Felder, die pfadbezogen sind (Vorwärtsschrägstriche, keine Laufwerksbuchstaben).
- Jeder SSH-Server kann einen unterschiedlichen Speicherort für sein Stammverzeichnis aufweisen. Zudem können die erlaubten Befehle, die relativ zu diesem Stammverzeichnis sind, variieren.

Operator "Datei komprimieren"

Der Operator "Datei komprimieren" komprimiert eine Datei oder ein Verzeichnis. In einer Windows-Umgebung wird er mithilfe des WINZIP-Befehlszeilenhilfsprogramms komprimiert. In UNIX-Umgebungen wird er mithilfe des Gzip-Hilfsprogramms komprimiert.

Voraussetzungen

- Das Befehlszeilenprogramm WZZIP muss auf dem Zielhost installiert werden, wenn dieser ein Windows-Host ist. Das Befehlszeilenprogramm WZZIP ist ein kostenloses Add-on für Anwender, die eine gültige Lizenz für die Standard- oder Pro-Version von WinZip 12 besitzen.
- Das Dienstprogramm Gzip wird für UNIX-Umgebungen benötigt.

Eingabeparameter

Quelldatei/-verzeichnisname

Gibt den Namen der Datei oder des Verzeichnisses an, die bzw. das komprimiert werden soll.

Hinweis: Auf dem Windows-Host ist die Erweiterung von komprimierten Dateien ".zip".

Wenn die Quelle ein Verzeichnis ist, wird auf einem UNIX-Host jede Datei in jenem Verzeichnis komprimiert und durch eine mit der Erweiterung ".gz" ersetzt.

Arbeitsverzeichnis

Gibt das Arbeitsverzeichnis zum Ausführen dieser Operation an.

Hinweis: Wenn das Arbeitsverzeichnis nicht angegeben wird, wird das Stammverzeichnis des Anwenders das Arbeitsverzeichnis sein.

Der Dateipfad kann absolut oder relativ zum Arbeitsverzeichnis sein.

Anwender-ID

Gibt das Anwenderkonto an, das während der Ausführung des Operators auf dem Host verwendet werden soll. Überschreibt den in den Eigenschaften der Operatorkategorieebene festgelegten Anwender.

Kennwort

Gibt das Kennwort für den Anwender an.

Ausgabeparameter

fileName

workingDir

userID

password

Operator "Datei kopieren"

Der Operator "Datei kopieren" kopiert die Quelle ins Ziel. Die Quelle und das Ziel können eine Datei oder ein Verzeichnis sein.

Eingabeparameter

Quelldatei/-verzeichnisname

Gibt die Datei oder das Verzeichnis an, in die bzw. das kopiert werden soll.

Zieldatei/-verzeichnis

Gibt die Datei oder das Verzeichnis an, in die bzw. das kopiert werden soll.

Arbeitsverzeichnis

Gibt das Arbeitsverzeichnis zum Ausführen dieser Operation an.

Hinweise: Wenn das Arbeitsverzeichnis nicht angegeben wird, fungiert das Stammverzeichnis des Anwenders als Arbeitsverzeichnis.

Der Dateipfad kann absolut oder relativ zum Arbeitsverzeichnis sein.

Anwender-ID

Gibt das Anwenderkonto an, das während der Ausführung des Operators auf dem Host verwendet werden soll. Überschreibt den auf Operatorkategorieebene festgelegten Anwender.

Kennwort

Gibt das Kennwort für den Anwender an.

Hinweise:

- Wenn das Ziel nicht existiert und die Quelle eine Datei ist, wird davon ausgegangen, dass das Ziel eine Datei ist.
- Wenn das Ziel nicht existiert und die Quelle ein Verzeichnis ist, wird davon ausgegangen, dass das Ziel ein Verzeichnis ist.
- Auf einem Windows-Host wird ein "cannot perform cyclic copy"-Fehler ausgegeben, wenn das Quellverzeichnis das Zielverzeichnis enthält.

Ausgabeparameter

fileName

destinationFileName

workingDir

userID

password

Operator "Ordner erstellen"

Der Operator "Ordner erstellen" gibt die Funktionalität an, um ein Verzeichnis zu erstellen. Der Operator löst keinen Fehler aus, wenn das Verzeichnis bereits vorhanden ist, und er erstellt Verzeichnisse bei Bedarf rekursiv.

Eingabeparameter

Verzeichnisname

Gibt das zu erstellende Verzeichnis an.

Standard-Shell

Berechtigungsmodi (nur für UNIX)

Legt den Berechtigungsmodus fest. Diese Eingabe ist nur für einen UNIX-Host gültig.

Arbeitsverzeichnis

Gibt das Arbeitsverzeichnis zum Ausführen dieser Operation an.

Hinweise: Wenn das Arbeitsverzeichnis nicht angegeben wird, fungiert das Stammverzeichnis des Anwenders als Arbeitsverzeichnis.

Der Dateipfad kann absolut oder relativ zum Arbeitsverzeichnis sein.

Anwender-ID

Gibt das Anwenderkonto an, das während der Ausführung des Operators auf dem Host verwendet werden soll. Überschreibt den in den Eigenschaften der Operatorkategorieebene angegebenen Anwender.

Kennwort

Gibt das Kennwort für den Anwender an.

Ausgabeparameter

fileName

defaultShell

Berechtigung

workingDir

userID

password

Operator "Datei dekomprimieren"

Der Operator "Datei dekomprimieren" extrahiert eine komprimierte Datei oder ein komprimiertes Verzeichnis.

Voraussetzungen

- Auf einem Windows-Host muss das WZZIP-Befehlszeilenprogramm auf dem Zielcomputer installiert sein. Das Befehlszeilenprogramm WZZIP ist ein kostenloses Add-on für Anwender, die eine gültige Lizenz für die Standard- oder Pro-Version von WinZip 12 besitzen.
- In UNIX-Umgebungen verwendet dieser Operator das Dienstprogramm "Gzip".

Eingabeparameter

Name der zu dekomprimierenden Datei/des zu dekomprimierenden Verzeichnisses

Gibt die Datei bzw. das Verzeichnis an, deren/dessen Komprimierung aufgehoben werden soll.

Arbeitsverzeichnis

Gibt das Arbeitsverzeichnis zum Ausführen dieser Operation an.

Hinweise: Wenn das Arbeitsverzeichnis nicht angegeben wird, fungiert das Stammverzeichnis des Anwenders als Arbeitsverzeichnis.

Der Dateipfad kann absolut oder relativ zum Arbeitsverzeichnis sein.

Anwender-ID

Gibt das Anwenderkonto an, das während der Ausführung des Operators auf dem Host verwendet werden soll. Überschreibt den in den Eigenschaften der Operatorkategorieebene angegebenen Anwender.

Kennwort

Gibt das Kennwort für den Anwender an.

Ausgabeparameter

compressFileName

workingDir

userID

password

Operator "Datei löschen"

Der Operator "Datei löschen" entfernt (löscht) eine Datei oder ein Verzeichnis.

Eingabeparameter

Quelldatei/-verzeichnisname

Gibt die Datei oder das Verzeichnis an, die bzw. das gelöscht werden soll.

Arbeitsverzeichnis

Gibt das Arbeitsverzeichnis zum Ausführen dieser Operation an.

Hinweis: Wenn das Arbeitsverzeichnis nicht angegeben wird, wird das Stammverzeichnis des Anwenders das Arbeitsverzeichnis sein.

Der Dateipfad kann absolut oder relativ zum Arbeitsverzeichnis sein.

Anwender-ID

Gibt das Anwenderkonto an, das während der Ausführung des Operators auf dem Host verwendet werden soll. Überschreibt den in den Eigenschaften der Operatorkategorieebene festgelegten Anwender.

Kennwort

Gibt das Kennwort für den Anwender an.

Hinweise:

- Beim Versuch, eine nicht vorhandene Datei oder ein nicht vorhandenes Verzeichnis zu löschen, wird ein Fehler ausgegeben.
- Beim Versuch, eine Datei oder ein Verzeichnis ohne ausreichende Berechtigungen zu löschen, wird ein Fehler ausgegeben.

Ausgabeparameter

fileName

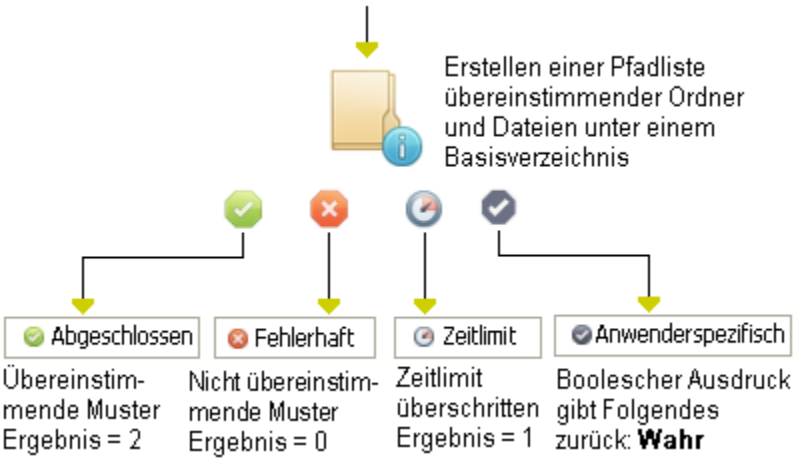
workingDir

userID

password

Operator "Verzeichnisinhalt abrufen"

Der Operator "Verzeichnisinhalt abrufen" erstellt eine Liste mit den Pfaden aller Ordner oder Dateien innerhalb eines angegebenen Verzeichnisses, die mit einer Suchbedingung übereinstimmen.


Erstellen einer Pfadliste übereinstimmender Ordner und Dateien unter einem Basisverzeichnis
Abgeschlossen
Übereinstim-mende Muster Ergebnis = 2
Fehlerhaft
Nicht übereinstim-mende Muster Ergebnis = 0
Zeitlimit
Zeitlimit überschritten Ergebnis = 1
Anwenderspezifisch
Boolescher Ausdruck gibt Folgendes zurück: Wahr

Eingabeparameter

Basisverzeichnis

Gibt den Pfad zum Verzeichnis an, in dem die Suche starten soll.

Dateipfad/Namensmaske

Gibt das vom Operator erkannte Muster an.

Musterübereinstimmung mit berücksichtigter Groß- bzw. Kleinschreibung

Wenn dies aktiviert ist, wird bei der Suche nach einem Muster die Groß- und Kleinschreibung berücksichtigt. Wenn dies deaktiviert ist, berücksichtigen die Buchstaben in einem Muster die Groß- und Kleinschreibung.

Elemente nach Zeitpunkt der letzten Änderung sortieren

Wenn dies aktiviert ist, werden die Ordner und Dateien in einem angegebenen Verzeichnis nach dem letzten Änderungsdatum sortiert.

Übereinstimmungsmuster für Datei-/Verzeichnisnamen

Wenn dies aktiviert ist, werden nur Datei- oder Verzeichnisnamen abgeglichen, nicht die Verzeichnisse innerhalb eines Pfads.

In Ergebnissen enthaltene Verzeichnisse

Wählen Sie eine der folgenden Optionen aus:

- Alle Verzeichnisse unter dem Basisverzeichnis
- Verzeichnisse, die übereinstimmende Dateien enthalten
- Verzeichnisse mit übereinstimmendem Pfad oder Namen

Rekursionsebene

Die Anzahl von Verzeichnisebenen, die bei der Übereinstimmung von Dateien oder Verzeichnissen absteigend durchsucht werden.

Statuszeitgeber (Sek.)

Das Intervall, während dessen die Bedingung mindestens zutreffen muss.

Ausgabeparameter

DirectoryCount

DirectoryList

FilesCount

FilesList

fileName

Muster

caseSensitive

timeSort

matchPath

dirFilterOptions

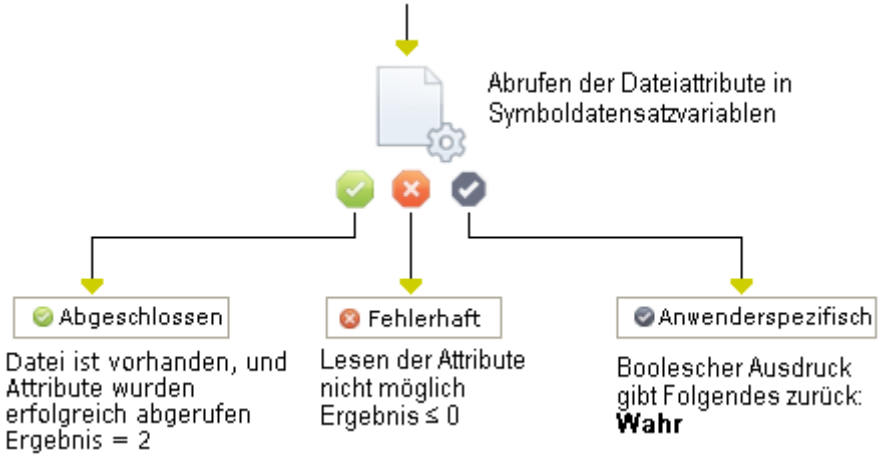
recursionLevel

stateTimer

timeOut

Operator "Dateiattribute abrufen"

Der Operator "Dateiattribute abrufen" liest die Attribute einer angegebenen Datei und speichert sie als Variablen (beispielsweise FileExists, FileSize, IsFile, IsLink) in seinem Operatordatensatz.



Um auf ein Attribut in einem CA Process Automation-Ausdruck zuzugreifen, verwenden Sie die folgende Syntax:

Process.Operator_name.field_name

Prozess greift auf den Prozessdatensatz zu.

Operator_name gibt den Operatordatensatz an.

field_name gibt das Attribut im Operatordatensatz an.

Eingabeparameter

Datei-/Verzeichnisname

Gibt den vollständigen Pfad für eine Datei oder ein Verzeichnis an. Der Speicherort muss für die Dateimanagement-Operatoren auf dem Kontaktpunkt zugänglich sein, wo es zur Laufzeit ausgeführt wird.

Beispiel:

`/tmp/IT PAM/scripts/backup_ora1.log`

Wenn Sie eine Datei oder ein Verzeichnis angeben, ohne den vollständigen Pfad anzugeben, verwenden die Dateimanagement-Operatoren das Verzeichnis "<*install_dir*>\server\c2o" als relativen Pfad für die angegebene Datei oder das angegebene Verzeichnis. In den meisten Fällen können Sie den Schrägstrich (/) in einer Pfadangabe verwenden.

Ausgabeparameter

@FileName

Der vollständige Pfad der Datei, der anhand des Ausdrucks "Dateinamen" berechnet wurde.

FileExists

1, wenn die Datei vorhanden ist; 0, wenn sie nicht vorhanden ist

FileName

FileTime

Gibt die Uhrzeit der Datei an.

FileDate

Gibt ein Dateidatum an.

IsDirectory

1, wenn ein Verzeichnis beschrieben, 0, wenn kein Verzeichnis beschrieben wird.

IsFile

1, wenn eine Datei beschrieben, 0, wenn keine Datei beschrieben wird.

isLink

1, wenn das Element eine symbolische Verknüpfung in UNIX ist, sonst 0.

FileOtherRead

FileOtherWrite

FilePermission

FileSize

Gibt die Größe der Datei in Byte an.

FileSizeKB

Die Größe der Datei in Kilobyte (1 KB = 1.024 Byte). Ein Bruchteil eines Kilobyte wird als ein Kilobyte betrachtet.

FileSizeMB

Die Größe der Datei in Megabyte (1 MB = 1.024 KB). Ein Bruchteil eines Megabyte wird als ein Megabyte betrachtet.

FileGroup

FileOwner

FileGroupExec

1, wenn die Gruppe ausführen kann, 0, wenn die Gruppe nicht ausführen kann.

FileGroupRead

1, wenn die Gruppe lesen kann, 0, wenn Gruppe nicht lesen kann.

FileGroupWrite

1, wenn die Gruppe schreiben kann, 0, wenn Gruppe nicht schreiben kann.

FileOtherExec

1, wenn andere ausführen können, 0, wenn andere nicht ausführen können

FileOwnerExec

1, wenn Verantwortliche ausführen können, 0, wenn Verantwortliche nicht ausführen können.

FileOwnerRead

1, wenn Verantwortliche lesen können, 0, wenn Verantwortliche nicht lesen können.

FileOwnerWrite

1, wenn Verantwortliche schreiben können, 0, wenn Verantwortliche nicht schreiben können.

IsSpecial

1, wenn eine besondere Systemdatei vorliegt, 0, wenn keine besondere Systemdatei vorliegt.

Hinweis: Die Definition einer besonderen Systemdatei ist plattformabhängig.

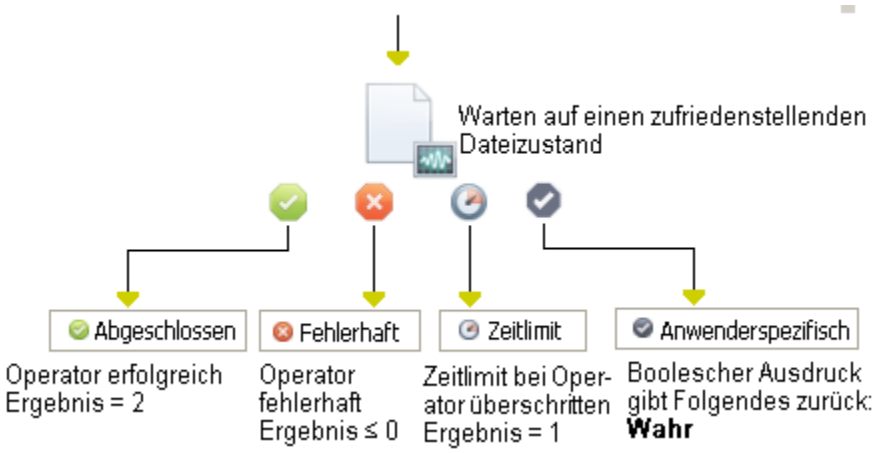
Beispiel

Auf UNIX-Plattformen kann dieser Operator einem Socket oder fifo entsprechen.

Andere (zum Beispiel FileOtherExec) bezieht sich auf Anwender, die weder als Verantwortlicher noch in der definierten Gruppe für die angegebene Berechtigung angegeben sind.

Operator "Datei überwachen"

Der Operator "Datei überwachen" gibt eine Wartezeit für eine Bedingung auf einer Datei oder einem Verzeichnis an. Dieser Operator ermöglicht es Ihnen, die Verarbeitung in einem Prozess für die Existenz oder Abwesenheit einer Datei oder eines Verzeichnisses zu verzögern.



Verfügbare Bedingungen:

- Die Existenz einer Datei, und optional, eine minimale Größe (in Byte).
- das Fehlen einer Datei
- Die Anwesenheit von Zeichenfolgen, die mit einem Muster in einer Datei übereinstimmen.

Ein Stabilitätszeitgeber gibt die minimale Verzögerung an, während derer die Bedingung kontinuierlich gehalten werden muss, bevor sie zurückgibt, dass der Operator erfolgreich abgeschlossen wurde. Zum Beispiel kann dieser Operator die minimale Größe einer Datei angeben, die mittels FTP-Übertragung versendet wird.

Tests für eine Datei sind diskret. Dieser Operator gibt nicht an, dass eine Bedingung kontinuierlich existiert. Der Operator "Datei überwachen" gibt vielmehr an, dass die Bedingung an jedem vom Operator ausgeführten Testintervall erfüllt wird.

Eingabeparameter

Datei-/Verzeichnisname

Gibt den vollständigen Pfad für eine zu überwachende Datei oder ein Verzeichnis an. Der Speicherort muss für die Dateimanagement-Operatoren auf dem Kontaktpunkt zugänglich sein, wo es zur Laufzeit ausgeführt wird.

Wenn Sie eine Datei oder ein Verzeichnis angeben, ohne den vollständigen Pfad anzugeben, verwenden die Dateimanagement-Operatoren das CA Process Automation Bin-Verzeichnis als relativen Pfad für die angegebene Datei oder das angegebene Verzeichnis. Normalerweise können Sie den Schrägstrich (/) in einer Pfadangabe verwenden.

Statuszeitgeber (Sek.)

Gibt die minimale Verzögerung in Sekunden an, während derer die angegebene Bedingung kontinuierlich gehalten werden muss, bevor CA Process Automation Verzweigungen für den Status "Abgeschlossen" ausführt.

Bedingung

Gibt die Bedingung an, die erfüllt werden muss, um die Verzweigungen für den Status "Abgeschlossen" auszuführen:

- Für "Vorhanden" muss die Datei existieren und die Dateigröße größer als der unter "Mindestdateigröße" festgelegte Wert sein.
- "Nicht vorhanden" erfordert, dass die Datei nicht vorhanden ist.
- "Musterübereinstimmung" gibt an, dass ein Muster von Zeichen in den Inhalten einer angegebenen Datei oder in den Namen von Dateien in einem angegebenen Verzeichnis auftreten.

Minimale Dateigröße

Wenn "Vorhanden" als Bedingung angegeben wird, gibt diese Option die minimale Größe in Byte an, die eine Datei erreichen muss, damit dieser Operator Verzweigungen für die Bedingung "Abgeschlossen" ausführt.

Muster

Wenn für Bedingung "Musterübereinstimmung" angegeben ist, gibt diese Option einen regulären Ausdruck an, der das Muster zurückgibt, nach dem der Operator sucht (siehe "Verwenden von Masken zur Angabe von Mustern in Zeichenfolgen" im *Handbuch für Inhaltsdesign*).

Um einer beliebigen Anzahl an Mehrfachzeilen zu entsprechen, können Sie im Feld "Struktur" das Maskierungszeichen \n verwenden. Im folgenden Beispiel werden Zeilen als Übereinstimmung gefunden, die mit "Protokoll" anfangen, danach eine beliebige Anzahl von neuen Zeilen enthalten und mit der Zeichenfolge mit dem Text "Error=89" enden:

```
"Log.*\n.*Error=89"
```

Wenn Sie bei der Programmierung mit Escape-Zeichen arbeiten, beachten Sie, dass dieses Escape-Zeichen \n für eine beliebige Anzahl an neuen Zeilen unter Windows oder UNIX steht. Dieses Escape-Zeichen steht nicht für ein einzelnes Zeilenvorschubzeichen.

Trennzeichen

Gibt das Zeichen an, mit dem die Zone unter "Struktur" begrenzt wird, die in den unter "Variablennamen" angegebenen Variablen gespeichert werden soll.

Starten am Ende der Datei

Beginnt mit der Suche am Ende einer angegebenen Datei, um das letzte Vorkommen eines Musters in einer Datei zu finden. Hiermit können Sie die neuesten Meldungen in einer Datei abgleichen.

Musterübereinstimmung mit berücksichtigter Groß- bzw. Kleinschreibung

Berücksichtigt bei der Suche nach einem Muster Groß- und Kleinbuchstaben. Wenn Sie dieses Kontrollkästchen nicht aktivieren, stimmen die Buchstaben in einem Muster sowohl mit Groß- als auch Kleinbuchstaben überein.

Offset für Dateisuche

Gibt eine Startposition für eine Suche an.

Um einen wiederholten Musterabgleich in einer Schleife durchzuführen, können Sie die Variablen "MatchPos" und "MatchEntry" im Operatordatensatz des Operators verwenden, um dort zu starten, wo die vorherige Suche nach Übereinstimmungen aufhörte:

```
Process.Operator_name. MatchedPos+ Len(Process.Operator_name.MatchedEntry)
```

Variablennamen

Gibt die Variablennamen an, in denen Text gespeichert wird, der mit den abgegrenzten Zonen im Muster übereinstimmt. Die Variablen "Operatordatensatz" sind über den Prozessdatensatz mittels des Schlüsselworts "Prozess" zugängig. Durch die Angabe der Variablennamen "Ebene" und "Code" würden beispielsweise extrahierte untergeordnete Zeichenfolgewerte den Operatordatensatz-Variablen "Process.Operator_name.Level" und "Process.Operator_name.Code" zugewiesen. Sie können die Variablen hinzufügen, entfernen und ordnen, mit denen übereinstimmende Zeichenfolgen über die Symbolleiste gespeichert werden.

Ausgabeparameter

LastRead POs

MatchedEnd

MatchedEntry

MatchedPos

fileName

stateTimer

Bedingung

minFileSize

Muster

Trennzeichen

startFromEnd

caseSensitive

fileSearchOffset

variableNames

Beispiel

Dieser Operator kann auf einen ausgehenden Operator warten, um eine Datei zu spoolen und zu löschen. Es würde ein Problem angezeigt werden, wenn eine Datei sich länger im Spoolverzeichnis befindet als eine angegebene Dauer. Der Prozess könnte dann einen Alarm ausführen, um einen Operator über das Problem zu informieren.

Operator "Lesen ab Datei"

Der Operator "Lesen ab Datei" liest den Inhalt der Datei in eine Datensatzvariable ein. Der Anwender kann auch bestimmte Zeilen lesen.

Die Datensatzvariable kann eine Zeichenfolge oder ein Zeichenfolgen-Array sein. Wenn die Datensatzvariable ein Array ist, kann es eine maximale Länge von 1024 haben. Zeilen, die dieses Limit überschreiten, werden ignoriert, und nach der Ausführung wird ein neues Datensatz-Variablenfeld namens "Warnungen" erstellt. Dieser neue Warnungsdatensatz enthält die Warnmeldung.

Eingabeparameter

Name der Quelldatei

Gibt die zu lesende Datei an.

Dateiinhalt in einem Zeichenfolgen-Array zurückgeben

Gibt an, ob die Variable "Datensatz" ein Zeichenfolgen-Array ist. Wenn dies aktiviert ist, wird davon ausgegangen, dass die Variable "Datensatz" eine Zeichenfolge ist.

Ab Zeilennummer

Gibt die Leitungsnummer in der Datei an, ab der der Dateiinhalt gelesen werden muss. Wenn dieses Feld leer gelassen wird, liest dieser Operator ab der Zeilennummer 1. Dieses Feld darf nur positive Werte ohne Null enthalten.

Bis Zeilennummer

Gibt die Leitungsnummer in der Datei an, bis wo der Inhalt gelesen werden muss. Dieses Feld darf nur positive Werte ohne Null enthalten. Wenn es leer gelassen wird, liest dieser Operator bis zum Dateiende.

Name der Datensatzvariablen

Gibt den Namen der Datensatzvariablen an.

Die Länge dieses Felds kann maximal 1024 sein. Zeilen, die dieses Limit überschreiten, werden ignoriert, und nach der Ausführung wird ein neues Datensatz-Variablenfeld "Warnungen" erstellt, das die Warnmeldung enthält.

Arbeitsverzeichnis

Gibt das Arbeitsverzeichnis zum Ausführen dieser Operation an.

Hinweise: Wenn das Arbeitsverzeichnis nicht angegeben wird, fungiert das Stammverzeichnis des Anwenders als Arbeitsverzeichnis.

Der Dateipfad kann absolut oder relativ zum Arbeitsverzeichnis sein.

Anwender-ID

Gibt das Anwenderkonto an, das während der Ausführung des Operators auf dem Host verwendet werden soll. Überschreibt den Anwender, der in der Konfiguration der Operatorkategorie festgelegt ist.

Kennwort

Gibt das Kennwort für die Anwender-ID an.

Dateicodierung

Gibt das Kodierungsschema an, das für das Lesen der Datei verwendet wird (UTF-8, UTF-16, US-ASCII, Windows-1250, Windows-1252, Shift_JIS).

Ausgabeparameter

DatasetVariable

Die Länge der Variablen "Datensatz" kann für ein Array maximal 1024 betragen. Zeilen, die dieses Limit überschreiten, werden ignoriert, und nach der Ausführung wird ein neues Datensatz-Variablenfeld "Warnungen" erstellt, das die Warnmeldung enthält.

fileName

isReturnAsStringArray

fromLineNumber

toLineNumber

datasetVariableName

workingDir

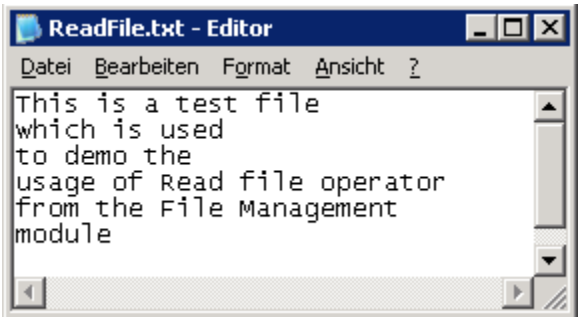
userID

password

Dateicodierung

Beispiel (Operator "Lesen aus Datei")

In diesem Beispiel können Sie Inhalt aus der Datei "ReadFile.txt" lesen. Der Dateiinhalt lautet folgendermaßen:



Dieses Beispiel berücksichtigt die folgenden Szenarien:

- Lesen aus der Datei
- Lesen aus der Datei und Speichern des Dateiinhalts als Array
- Lesen von bestimmten Zeilen aus der Datei
- Lesen aus der Datei und Speichern des Dateiinhalts in einem Datensatz

Gehen Sie folgendermaßen vor:

1. Entwerfen Sie einen Prozess mit dem Operator *Lesen aus Datei*, wie in der folgenden Abbildung dargestellt:

2. Doppelklicken Sie auf den Operator *Lesen aus Datei,* um die Eigenschaften "Lesen aus Datei" zu öffnen, und wählen Sie den Bereich *Datei in Datensatzvariable einlesen* aus.

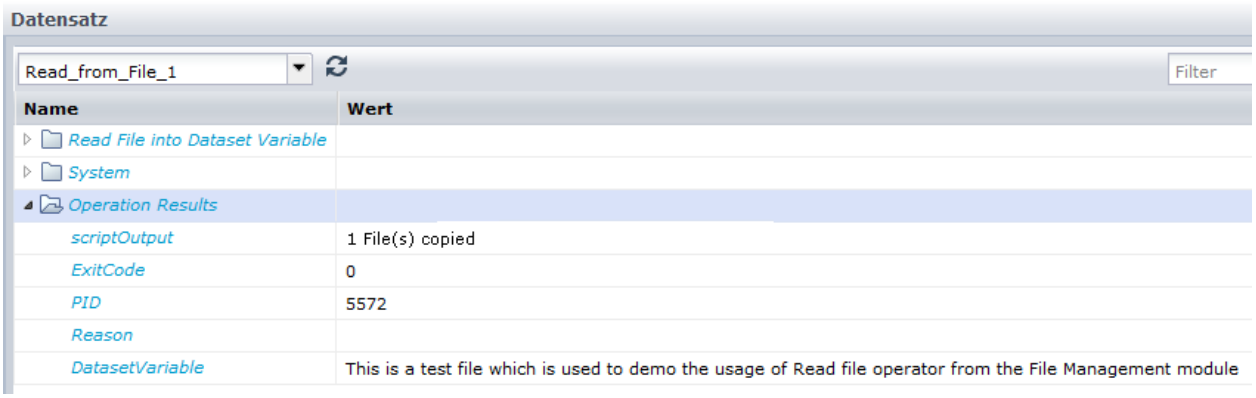
   So lesen Sie aus der Datei

3. Geben Sie den folgenden Dateipfad im Feld "Name der Quelldatei" ein.

   (Windows) C:\\ReadFile.txt

   (UNIX) /root/readfromfile.txt

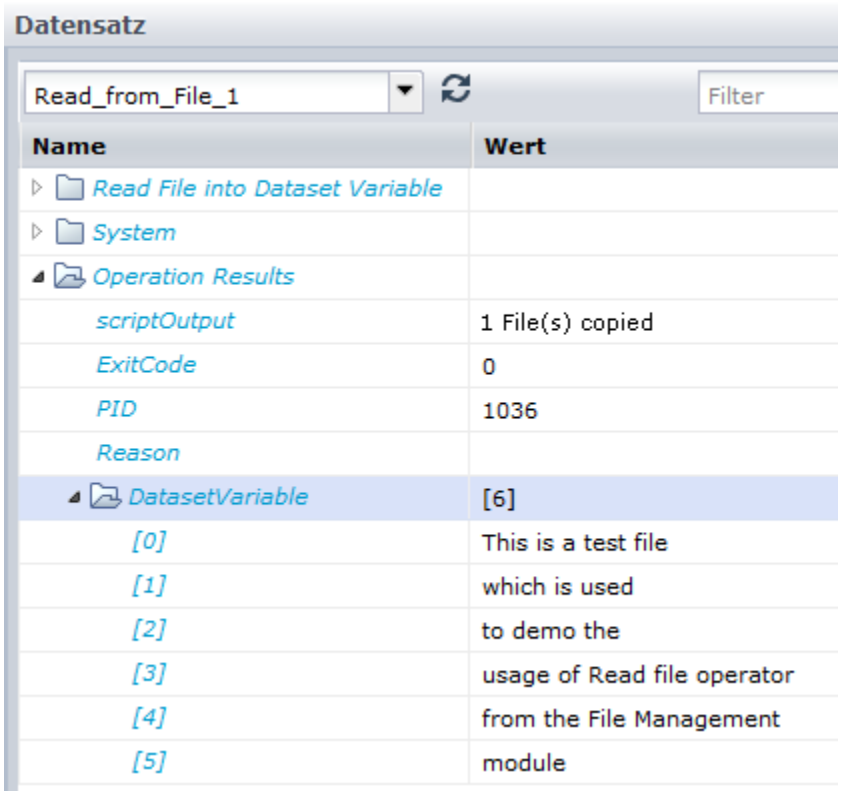
4. Führen Sie den Prozess aus.
5. Öffnen Sie die Betriebsablauf-Ergebnisse, um den DatasetVariable-Wert anzuzeigen, wie in der folgenden Abbildung dargestellt:



So lesen Sie aus der Datei und speichern den Dateiinhalt als Array

6. Aktivieren Sie das Kontrollkästchen *Dateiinhalt in einem Zeichenfolgen-Array zurückgeben*.

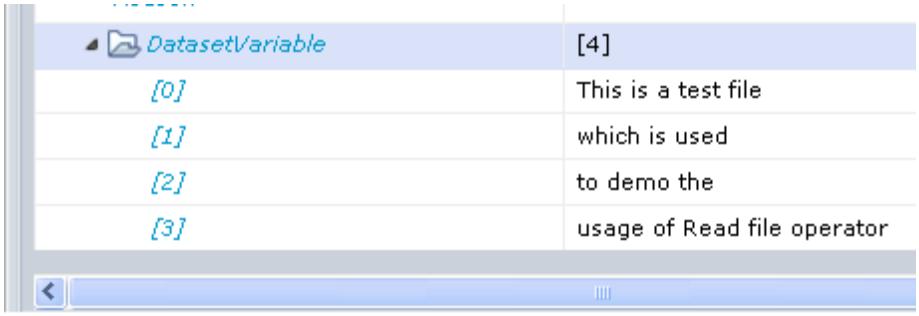
   Der Dateiinhalt ist als Array verfügbar, wie in der folgenden Abbildung dargestellt:



So lesen Sie bestimmte Zeilen in der Datei

7. Geben Sie in das Feld "Ab Zeilennummer" 1 ein, um in der Datei "ReadFile.txt" ab der ersten Zeile zu lesen.
8. Geben Sie 4 in das Feld "Bis Zeilennummer" ein, um in der Datei "ReadFile.txt" bis zur vierten Zeile zu lesen.

   Der Dateiinhalt wird von der Zeile 1 bis zur Zeile 4 gelesen und im Datensatz gespeichert, wie in der folgenden Abbildung dargestellt:

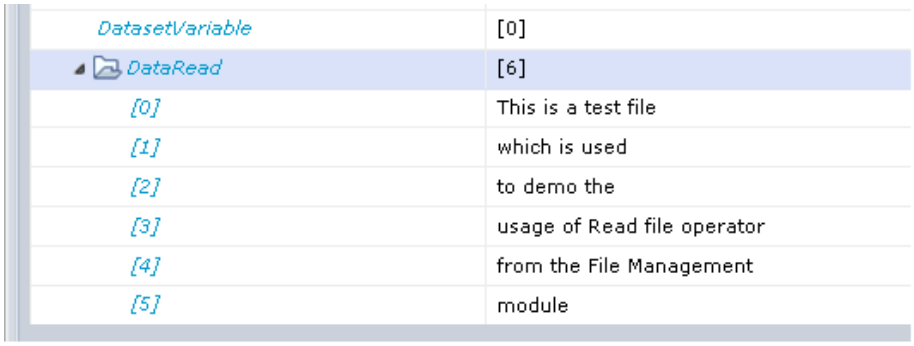


DatasetVariable	[4]
[0]	This is a test file
[1]	which is used
[2]	to demo the
[3]	usage of Read file operator

So lesen Sie aus der Datei und speichern den Dateiinhalt in einer Datensatzvariable

9. Geben Sie als Namen der Datensatzvariable "DataRead" ein.

   Der Dateiinhalt wird im Datensatz "DataRead" und nicht im Systemdatensatz gespeichert, wie in der folgenden Abbildung dargestellt:



DatasetVariable	[0]
DataRead	[6]
[0]	This is a test file
[1]	which is used
[2]	to demo the
[3]	usage of Read file operator
[4]	from the File Management
[5]	module

Sie können auch die Anwender-ID und das Kennwort des Anwenderkontos angeben, das die Berechtigung zum Ausführen eines Prozesses erhalten soll. Wenn Sie Anwender-ID- und Kennwortwerte auf Operatorebene angeben, überschreiben Sie die Werte, die im Feld "Anmeldeinformationen anfordern" definiert sind. Sie definieren das Feld "Anmeldeinformationen anfordern" in den Eigenschaften "Dateimanagement".

Hinweis: Stellen Sie sicher, dass Sie dem Anwender eine *Leseberechtigung* und für das PAM-Installationsverzeichnis eine Berechtigung zum *Lesen und Ausführen* erteilen, um die ReadFile-Service-Operation auszuführen. Weitere Informationen zum Konfigurieren des Dateimanagements finden Sie im *Handbuch für Inhaltsadministratoren* im Abschnitt "Konfigurieren des Dateimanagements".

Operator "Datei umbenennen"

Der Operator "Datei umbenennen" stellt Funktionalität zum Umbenennen einer Datei oder eines Verzeichnisses bereit.

Eingabeparameter

Quelldatei/-verzeichnisname

Gibt die bzw. das umzubenennende Datei/Verzeichnis an.

Neuer Datei-/Verzeichnisname

Gibt den neuen Namen für die Datei bzw. das Verzeichnis an.

Arbeitsverzeichnis

Gibt das Arbeitsverzeichnis zum Ausführen des Operators an.

Hinweise: Wenn das Arbeitsverzeichnis nicht angegeben wird, fungiert das Stammverzeichnis des Anwenders als Arbeitsverzeichnis.

Der Dateipfad kann absolut oder relativ zum Arbeitsverzeichnis sein.

Anwender-ID

Gibt das Anwenderkonto an, das während der Ausführung des Operators auf dem Host verwendet werden soll. Überschreibt den in den Parametern der Operatorkategorieebene angegebenen Anwender.

Kennwort

Gibt das Kennwort für die Anwender-ID an.

Hinweis: Für UNIX wird die Datei, wenn der Zielspeicherort nicht die Quelle ist, an den Zielspeicherort verschoben.

Ausgabeparameter

fileName

newFileName

workingDir

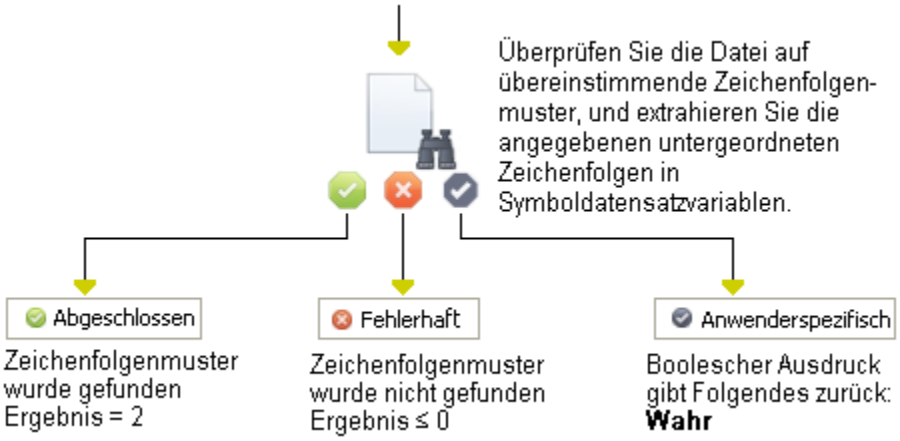
userID

password

Operator "Dateiinhalt durchsuchen"

Der Operator "Dateiinhalt durchsuchen" extrahiert Informationen aus relativ kleinen Dateien. Dieser Operator durchsucht den Inhalt einer Textdatei oder eines Verzeichnisses nach einer Zeichenfolge, die mit einem angegebenen Muster übereinstimmt. Das Muster kann Teilzeichenfolgen angeben, die in Variablen eines Operatordatensatzes extrahiert werden sollen. Trennzeichen im Muster können die Bereiche angeben, die in die Variablen extrahiert werden sollen.

- Bei einer Datei durchsucht dieser Operator die Inhalte der Datei nach dem angegebenen Muster.
- Eine Gruppensuche wird empfohlen, um große Dateien, die Gigabytes von Information enthalten, effizient zu durchsuchen.



Eingabeparameter

Datei-/Verzeichnisname

Gibt den Pfad für die Datei oder das Verzeichnis an, die für Musterübereinstimmungen durchsucht werden sollen. Der Pfad muss für die Dateimanagement-Operatoren auf dem Kontaktpunkt zugänglich sein, wo es ausgeführt wird, zugänglich sein. Bei einer Datei durchsucht der Operator die Inhalte der Datei nach dem angegebenen Muster.

Wenn Sie eine Datei oder ein Verzeichnis angeben, ohne den vollständigen Pfad anzugeben, verwenden die Dateimanagement-Operatoren das Verzeichnis "<*install_dir*>\server\c2o" als relativen Pfad für die angegebene Datei oder das angegebene Verzeichnis. In den meisten Fällen können Sie den Schrägstrich (/) in einer Pfadangabe verwenden.

Muster

Gibt das Muster an, wonach der Operator sucht (siehe "Verwenden von Masken zur Angabe von Mustern in Zeichenfolgen" im *Handbuch für Inhaltsdesign*).

Das Muster verwendet das Nummernsymbol (#) als Trennzeichen, um Werte für die Variablen "Ebene" und "Code" zurückzugeben:

```
"BACKUP LEVEL #.*# - CODE #.*#"
```

Bei der Zeichenfolge "BACKUP LEVEL A400 - CODE FSC137.0359" würde dieses Muster die untergeordneten Zeichenfolgen "A400" der Variablen "Ebene" und "FSC731.0359" der Variablen "Code" zuweisen. Die Zuweisung erfolgt in der gleichen Reihenfolge, in der die Variablen unter "Variablennamen" definiert wurden.

Um einer beliebigen Anzahl an Mehrfachzeilen zu entsprechen, können Sie im Feld "Struktur" das Maskierungszeichen \n verwenden. Im folgenden Beispiel werden Zeilen als Übereinstimmung gefunden, die mit "Protokoll" anfangen, danach eine beliebige Anzahl von neuen Zeilen enthalten und mit der Zeichenfolge mit dem Text "Error=89" enden:

```
"Log.*\n.*Error=89"
```

Wenn Sie bei der Programmierung mit Escape-Zeichen arbeiten, beachten Sie, dass dieses Escape-Zeichen \n für eine beliebige Anzahl an neuen Zeilen unter Windows oder UNIX steht. Dieses Escape-Zeichen steht nicht für ein einzelnes Zeilenvorschubzeichen.

Trennzeichen

Gibt ein Zeichen an, mit dem die Zone begrenzt wird, die in den von "Variablennamen" angegebenen Variablen gespeichert werden soll, beispielsweise das #-Symbol.

Starten am Ende der Datei

Aktivieren Sie dieses Kontrollkästchen, um die Suche am Ende einer angegebenen Datei zu beginnen. Diese Option wird verwendet, um das letzte Vorkommen eines Musters in einer Datei zu suchen. Hiermit können Sie die neuesten Meldungen in einer Datei abgleichen.

Musterübereinstimmung mit berücksichtigter Groß- bzw. Kleinschreibung

Aktivieren Sie dieses Kontrollkästchen, um bei der Suche nach einem Muster Groß- und Kleinbuchstaben zu berücksichtigen. Wenn Sie dieses Kontrollkästchen nicht aktivieren, stimmen die Buchstaben in einem Muster sowohl mit Groß- als auch Kleinbuchstaben überein.

Offset für Dateisuche

Gibt eine Startposition für eine Suche an. Der Wert steht für die Anzahl von Zeichen am Anfang der Datei, es sei denn, das Kontrollkästchen "Starten am Ende der Datei" wurde aktiviert. In diesem Fall repräsentiert der Wert die Anzahl von Zeichen vom Ende der Datei.

Um einen wiederholten Musterabgleich in einer Schleife durchzuführen, können Sie die Variablen "MatchPos" und "MatchEntry" im Operatordatensatz eines früheren Operators "Dateiinhalt durchsuchen" verwenden, um dort zu starten, wo die vorherige Suche nach Übereinstimmungen aufhörte:

```
Process.Operator_name.MatchPos + Len(Process.Operator_name.MatchEntry)
```

Variablennamen

Gibt die Variablennamen an, in denen Text gespeichert wird, der mit den abgegrenzten Zonen im Muster übereinstimmt. Abgegrenzte Zonen werden in der in der Variablenliste angegebenen Reihenfolge in den aufgelisteten Variablen gespeichert. Die Variablen "Operatordatensatz" sind über den Prozessdatensatz mittels des Schlüsselworts "Prozess" zugängig.

Zum Beispiel durch Angeben der Variablennamen "Ebene" und "Code", um die extrahierten untergeordneten Zeichenfolgewerte den Variablen des Operatordatensatzes "Process.Operator_name.Level" und "Process.Operator_name.Code" zuzuweisen.

Sie können die Variablen hinzufügen, entfernen und ordnen, mit denen übereinstimmende Zeichenfolgen über die Symbolleiste gespeichert werden.

Ausgabeparameter

fileName

Muster

Trennzeichen

startFromEnd

caseSensitive

fileSearchOffset

variableNames

LastReadPos

MatchedEnd

MatchedEntry

MatchedPos

Operator "Dateieigentümer aktualisieren"

Der Operator "Dateieigentümer aktualisieren" ändert den Anwender und/oder die Gruppen, der bzw. als Verantwortlicher für eine Datei fungiert. Nur ein Superuser kann den Verantwortlichen und die Gruppe ändern, zu denen eine Datei gehört. Dieser Operator wird nur auf einem UNIX-Host unterstützt.

Eingabeparameter

Anwendername

Gibt den Verantwortlichen der Datei an.

Gruppenname

Gibt die Gruppe an, der die Datei angehört.

Hinweis: Der Anwender muss in mindestens einem der Felder "Anwendername" oder "Gruppenname" eine Eingabe machen.

Quelldatei/-verzeichnisname

Gibt den Namen der Datei oder des Verzeichnisses an, dessen Verantwortlicher geändert werden muss.

Rekursiv

Wenn dies aktiviert ist, gibt dies an, dass der Dateieigentümer rekursiv geändert wird.

Arbeitsverzeichnis

Gibt das Arbeitsverzeichnis zum Ausführen dieser Operation an.

Hinweise: Wenn das Arbeitsverzeichnis nicht angegeben wird, fungiert das Stammverzeichnis des Anwenders als Arbeitsverzeichnis.

Der Dateipfad kann absolut oder relativ zum Arbeitsverzeichnis sein.

Anwender-ID

Gibt das Anwenderkonto an, das während der Ausführung des Operators auf dem Host verwendet werden soll. Dieses Feld überschreibt den in den Eigenschaften der Operatorkategorieebene angegebenen Anwender.

Kennwort

Gibt das Kennwort für die Anwender-ID an.

Ausgabeparameter

userName

groupName

fileName

isRecursive

workingDir

userID

password

Operator "Dateiberechtigung aktualisieren"

Der Operator "Dateiberechtigung ändern" ändert die Berechtigungen einer gegebenen Datei entsprechend dem Modus, wobei es sich um eine Oktalzahl handeln kann, die das Bitmuster für die neuen Berechtigungen darstellt, oder um eine symbolische Darstellung der vorzunehmenden Änderungen (+-= rwxXstugoa). Dieser Operator wird nur auf einem UNIX-Host unterstützt.

Eingabeparameter

Quelldatei/-verzeichnisname

Gibt den Namen der Datei oder des Verzeichnisses an, deren bzw. dessen Berechtigung sich ändert.

Berechtigung (Modi)

Gibt Berechtigungen oder Modus für die Datei an, wobei es sich um eine Oktalzahl handeln kann, die das Bitmuster für die neuen Berechtigungen darstellt, oder um eine symbolische Darstellung der vorzunehmenden Änderungen (+-= rwxXstugoa).

Rekursiv

Gibt an, dass Dateien und Verzeichnisse rekursiv geändert werden.

Arbeitsverzeichnis

Gibt das Arbeitsverzeichnis zum Ausführen dieses Operators an.

Hinweise: Wenn das Arbeitsverzeichnis nicht angegeben wird, fungiert das Stammverzeichnis des Anwenders als Arbeitsverzeichnis.

Der Dateipfad kann absolut oder relativ zum Arbeitsverzeichnis sein.

Anwender-ID

Gibt das Anwenderkonto an, das während der Ausführung des Operators auf dem Host verwendet werden soll. Überschreibt den in den Eigenschaften der Operatorkategorieebene festgelegten Anwender.

Kennwort

Gibt das Kennwort für die Anwender-ID an.

Ausgabeparameter

fileName

Berechtigung

isRecursive

workingDir

userID

password

Operator "Dateizeitstempel aktualisieren"

Die Operator "Dateizeitstempel aktualisieren" ändert den Zeitstempel von Dateien, beispielsweise aktualisiert er die Zugriffs- und Änderungszeit der einzelnen Dateien mit der aktuellen Uhrzeit oder einem vom Anwender angegebenen Zeitstempel. Dieser Operator wird nur auf einem UNIX-Host unterstützt.

Eingabeparameter

Quelldatei/-verzeichnisname

Gibt den Namen der Datei oder des Verzeichnisses an, deren bzw. dessen Zeitstempel sich ändert.

Zeitstempel ([[JJ]JJ]MMTThhmm[.ss]):

Verwenden Sie dieses Feld im Format [[JJ]JJ]MMTThhmm[.ss] anstelle der aktuellen Zeit.

Wobei die Buchstaben für Folgendes stehen:

CC

Gibt die ersten zwei Ziffern des Jahres an (das Jahrhundert).

YY

Gibt die zweiten zwei Ziffern des Jahres an. Wenn "JJ" ohne Jahrhundertangabe angegeben wird, wird bei einem "JJ"-Wert zwischen 69 und 99 ein Jahrhundertwert von 19 ausgegeben. Sonst wird ein Jahrhundertwert von 20 verwendet.

Wenn die Buchstabenpaare "JJ" (Jahrhundert) und "JJ" nicht angegeben werden, wird automatisch das aktuelle Jahr verwendet.

MM

Gibt den Monat des Jahres an, von 1 bis 12.

DD

Gibt den Tag des Monats an, von 1 bis 31.

HH

Gibt die Stunde des Tages an, von 0 bis 23.

mm

Gibt die Minute der Stunde an, von 0 bis 59.

ss

Die Sekunde der Minute, von 0 bis 61.

Wenn das Buchstabenpaar "ss" nicht angegeben wird, wird automatisch der Wert 0 verwendet.

Zugriffszeit ändern

Gibt an, dass die Zugriffszeit geändert werden soll. Die Option ist standardmäßig aktiviert.

Änderungszeitpunkt ändern

Gibt an, dass die Änderungszeit geändert werden soll. Die Option ist standardmäßig aktiviert.

Arbeitsverzeichnis

Gibt das Arbeitsverzeichnis an, in dem dieser Operator ausgeführt werden soll.

Hinweis: Wenn das Arbeitsverzeichnis nicht angegeben wird, wird das Stammverzeichnis des Anwenders das Arbeitsverzeichnis sein.

Der Dateipfad kann absolut oder relativ zum Arbeitsverzeichnis sein.

Anwender-ID

Gibt das Anwenderkonto an, das während der Ausführung des Operators auf dem Host verwendet werden soll. Überschreibt den in den Eigenschaften der Operatorkategorieebene angegebenen Anwender.

Kennwort

Gibt das Kennwort für die Anwender-ID an.

Ausgabeparameter

fileName

timeStamp

isChangeAccessTime

isChangeModificationTime

workingDir

userID

password

Operator "Datei schreiben"

Der Operator "Datei schreiben" schreibt den Inhalt der Datensatzvariablen in eine Datei. Die Datensatzvariable kann eine Zeichenfolge oder ein Zeichenfolgen-Array sein. Dieser Operator gibt auch eine Option, um die Inhalte zu überschreiben oder an eine vorhandene Datei anzuhängen.

Eingabeparameter

Dateiinhalt als Array

Wenn dies aktiviert ist, gibt es an, dass die Datensatzvariable ein Zeichenfolgen-Array ist.

Dateiinhalt

Wenn Sie das Kontrollkästchen "Dateiinhalt als Array" nicht aktivieren, speichern Sie die Inhalte in diesem Feld in der angegebenen Datei ab.

Dateiinhalt als Array

Wenn das Kontrollkästchen "Dateiinhalt als Array" aktiviert ist, werden die Inhalte in diesem Feld in die angegebene Datei geschrieben, wobei jeder Index als eine neue Linie in der Datei angegeben wird.

Sie können Variablen hinzufügen, entfernen und sie mithilfe der Symbolleiste ordnen.

Zieldateiname

Gibt den Namen der Zieldatei an.

Hinweis: Wenn die Zieldatei nicht existiert, wird die Datei erstellt, bevor Daten in der Datei abgespeichert werden.

Anhängen (bei vorhandener Datei)

Gibt an, dass der Inhalt der Datensatzvariable an eine Datei angehängt ist, wenn sie bereits vorhanden ist. Die Option ist standardmäßig aktiviert.

Arbeitsverzeichnis

Gibt das Arbeitsverzeichnis zum Ausführen dieses Operators an.

Hinweise: Wenn das Arbeitsverzeichnis nicht angegeben wird, fungiert das Stammverzeichnis des Anwenders als Arbeitsverzeichnis.

Der Dateipfad kann absolut oder relativ zum Arbeitsverzeichnis sein.

Anwender-ID

Gibt das Anwenderkonto an, das während der Ausführung des Operators auf dem Host verwendet werden soll. Überschreibt den in den Parametern der Operatorkategorieebene angegebenen Anwender.

Kennwort

Gibt das Kennwort für die Anwender-ID an.

Dateicodierung

Gibt das Kodierungsschema an, das für das Schreiben in der Datei verwendet wird (UTF-8, UTF-16, US-ASCII, Windows-1250, Windows-1252, Shift_JIS).

Ausgabeparameter

fileContentsMode

fileContents

fileContentsAsArray

fileName

isFileAppend

workingDir

userID

password

Dateicodierung

Kapitel 10: Dateiübertragung

Die Dateitransfer-Operatoren geben Dateiübertragungsoperatoren an (FTP/SFTP).

Verwenden Sie die Dateitransfer-Operatoren, um Verzeichnisse und Dateien, beispielsweise FTP- oder einen SFTP-Client, zu verwalten. Diese Operatoren stellen Verbindungen mit standardmäßigen FTP-Servern auf Zielcomputern her. Der Remote-Host für alle Dateiübertragungsoperatoren muss über einen konfigurierten FTP-Server verfügen.

Wichtig! Abgesehen von den Operatoren "Datei über TFTP herunterladen" und "Datei über TFTP hochladen" gelten die folgenden Bedingungen für alle Operatoren in dieser Kategorie, wenn sie auf einem Remote-Windows-Host über einen Proxy-Kontaktpunkt ausgeführt werden:

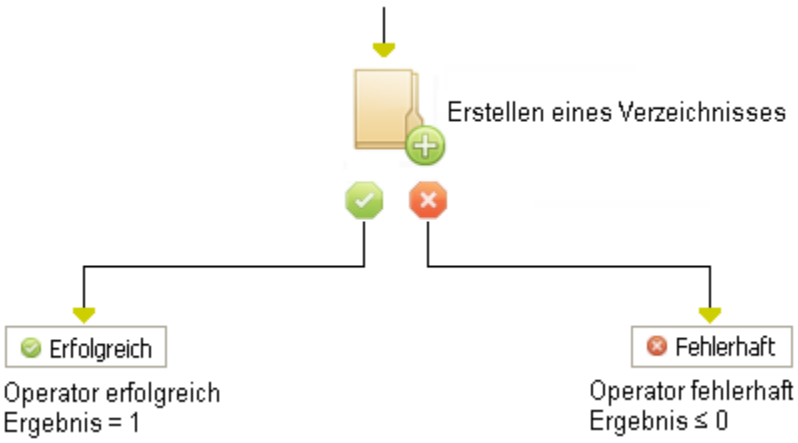
- Verwenden Sie Pfade im UNIX-Stil für Felder, die pfadbezogen sind (Vorwärtsschrägstriche, keine Laufwerksbuchstaben).

- Jeder SSH-Server kann einen unterschiedlichen Speicherort für sein Stammverzeichnis aufweisen. Zudem können die erlaubten Befehle, die relativ zu diesem Stammverzeichnis sind, variieren.

Operator "Verzeichnis erstellen"

Der Operator "Verzeichnis erstellen" erstellt ein Verzeichnis im Remote-Dateisystem.

Damit der Operator ein Verzeichnis erstellen kann, müssen die angegebenen Anwender-Anmeldeinformationen über die entsprechenden Berechtigungen verfügen, um auf dem Remote-Host das Verzeichnis ändern und Dateien schreiben zu können. Der Remote-Host muss über einen konfigurierten FTP-Server verfügen.



Eingabeparameter

Remote-Pfad

Gibt den Pfad für das Verzeichnis an, das auf dem Remote-Host erstellt werden soll. Zum Beispiel:

```
/temp/IT PAM/scripts.
```

Das übergeordnete Verzeichnis ("/temp/IT PAM" im Beispiel) muss vorhanden sein, damit der Operator "Dateitransfer" diesen Operator erfolgreich abschließt. In diesem Feld kann auch der relative Pfad angegeben werden. Der relative Pfad bezieht sich dann auf das Stammverzeichnis des FTP-Anwenders.

Remote-Host

Gibt die IP-Adresse oder FTP-URL für den Remote-Host an. Für eine FTP-Site auf dem Intranet Ihres Unternehmens können Sie den Servernamen (//Servername) für die FTP-Site angeben.

Standardmäßig verwenden FTP-Sites Port 21. Sie können FTP-Sites auch private Ports zuweisen. Private Ports liegen zwischen 49152 und 65535. Um einen privaten FTP-Port anzugeben, fügen Sie am Ende der Adresse einen Doppelpunkt (:) und im Anschluss eine Nummer eines privaten Ports an. Auf die folgende Weise wird Port 50021 für einen Remote-FTP-Server angegeben:

```
172.24.36.107:50021
```

Remote-Anwender-ID

Gibt eine Anwender-ID an, um auf den Remote-FTP-Host zuzugreifen.

Remote-Anwenderkennwort

Gibt das Kennwort an, um auf den Remote-FTP-Host zuzugreifen.

Sicheres Dateiübertragungsverfahren (SFTP) verwenden

Öffnet eine sichere FTP (SFTP)-Sitzung. SFTP ähnelt FTP, aber im Gegensatz zu FTP wird die gesamte Sitzung verschlüsselt. Es werden keine Kennwörter in reiner Textform gesendet, wodurch SFTP viel weniger anfällig für Abhörangriffe durch Drittparteien ist.

Geben Sie einen optionalen SITE-Befehl an

Gibt an, dass SITE-Befehl verwendet wird, um Services aufzurufen, die dem Hostsystem eigen sind. Verwenden Sie dann das Feld "Standortparameter", um einen Ausdruck anzugeben, der Parameter für den SITE-Befehl zurückgibt.

Site-Parameter

Geben Sie einen CA Process Automation-Ausdruck an, der Parameter für den SITE-Befehl zurückgibt.

Ausgabeparameter

remotedir

remoteHost

remoteUserId

remoteUserPassword

secureFtp

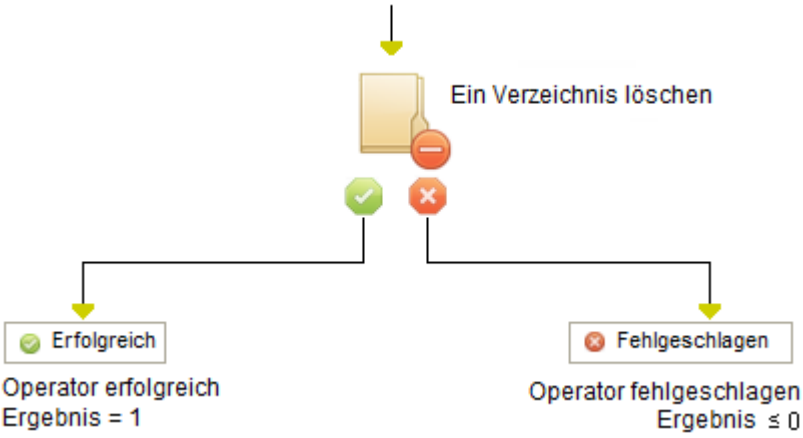
siteCommand

siteParameters

Operator "Verzeichnis löschen"

Der Operator "Verzeichnis löschen" löscht ein angegebenes Verzeichnis aus dem Remote-Dateisystem. Das angegebene Verzeichnis muss leer sein, damit der Prozess diesen Operator ausführt.

Damit ein Verzeichnis gelöscht werden kann, müssen die angegebenen Anwenderanmeldeinformationen über die entsprechenden Berechtigungen zum Ändern des Verzeichnisses und Schreiben auf dem Remote-Host verfügen. Der Remote-Host muss über einen konfigurierten FTP-Server verfügen.



Eigenschaften von "Remote-Verzeichnis löschen"

Remote-Pfad

Gibt den Pfad für das Verzeichnis an, das auf dem Remote-Host gelöscht werden soll. Beispiel: /temp/IT PAM/scripts. Das angegebene Verzeichnis muss leer sein, damit die Dateitransfer-Operatoren diesen Operator abschließen.

Remote-Host

Gibt die IP-Adresse oder FTP-URL für den Remote-Host an. Für eine FTP-Site auf dem Intranet Ihres Unternehmens können Sie den Servernamen (//Servername) für die FTP-Site angeben.

Standardmäßig verwenden FTP-Sites Port 21. Sie können FTP-Sites auch private Ports zuweisen. Private Ports liegen zwischen 49152 und 65535. Um einen privaten FTP-Port anzugeben, fügen Sie am Ende der Adresse einen Doppelpunkt (:) und im Anschluss eine Nummer eines privaten Ports an. Auf die folgende Weise wird Port 50021 für einen Remote-FTP-Server angegeben:

```
172.24.36.107:50021
```

Remote-Anwender-ID

Gibt eine Anwender-ID an, um auf den Remote-FTP-Host zuzugreifen.

Remote-Anwenderkennwort

Gibt das Kennwort an, um auf den Remote-FTP-Host zuzugreifen.

Sicheres Dateiübertragungsverfahren (SFTP) verwenden

Öffnet eine sichere FTP (SFTP)-Sitzung. SFTP ähnelt FTP, aber im Gegensatz zu FTP wird die gesamte Sitzung verschlüsselt. Es werden keine Kennwörter in reiner Textform gesendet, wodurch SFTP viel weniger anfällig für Abhörangriffe durch Drittparteien ist.

Geben Sie einen optionalen SITE-Befehl an

Gibt an, dass SITE-Befehl verwendet wird, um Services aufzurufen, die dem Hostsystem eigen sind. Verwenden Sie dann das Feld "Standortparameter", um einen CA Process Automation-Ausdruck anzugeben, der Parameter für den SITE-Befehl zurückgibt.

Alle Dateien/Unterverzeichnisse aus dem Zielverzeichnis entfernen

Diese Option wird verwendet, um ein Verzeichnis zu löschen, das nicht leer ist. Wenn dieses Kontrollkästchen ausgewählt ist, werden alle Verzeichnisse unter dem angegebenen Verzeichnis einschließlich des angegebenen Verzeichnisses gelöscht. Wenn dieses Kontrollkästchen deaktiviert ist, führt jeder Versuch, ein nicht leeres Verzeichnis zu löschen, zu einem Serviceoperator-Fehler.

Site-Parameter

Geben Sie einen CA Process Automation-Ausdruck an, der Parameter für den SITE-Befehl zurückgibt.

Ausgabeparameter

remotedir

remoteHost

remoteUserId

remoteUserPassword

secureFtp

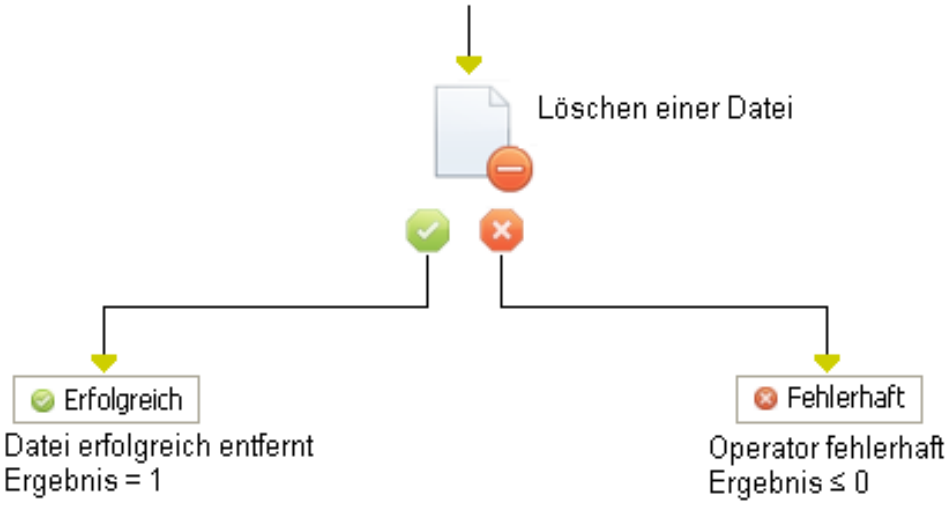
siteCommand

forceDelete

siteParameters

Operator "Datei löschen"

Der Operator "Datei löschen" entfernt eine angegebene Datei von einem Remote-Computer. Der Remote-Host muss über einen konfigurierten FTP-Server verfügen. Damit der Operator eine Datei löschen kann, müssen die angegebenen Anwender-Anmeldeinformationen über die entsprechenden Berechtigungen verfügen, um auf dem Remote-Host das Verzeichnis ändern und Dateien schreiben zu können.



Eingabeparameter

Remote-Pfad

Gibt den Pfad für die Datei an, die auf dem Remote-Host gelöscht werden soll. Beispiel: /tmp/IT PAM/scripts/script_ora1.sh.

Remote-Host

Gibt die IP-Adresse oder FTP-URL für den Remote-Host an. Für eine FTP-Site auf dem Intranet Ihres Unternehmens können Sie den Servernamen (//Servername) für die FTP-Site angeben.

Standardmäßig verwenden FTP-Sites Port 21. Sie können FTP-Sites auch private Ports zuweisen. Private Ports liegen zwischen 49152 und 65535. Um einen privaten FTP-Port anzugeben, fügen Sie am Ende der Adresse einen Doppelpunkt (:) und im Anschluss eine Nummer eines privaten Ports an. Auf die folgende Weise wird Port 50021 für einen Remote-FTP-Server angegeben:

```
172.24.36.107:50021
```

Remote-Anwender-ID

Gibt eine Anwender-ID an, um auf den Remote-FTP-Host zuzugreifen.

Remote-Anwenderkennwort

Gibt das Kennwort an, um auf den Remote-FTP-Host zuzugreifen.

Sicheres Dateiübertragungsverfahren (SFTP) verwenden

Öffnet eine sichere FTP (SFTP)-Sitzung. SFTP ähnelt FTP, aber im Gegensatz zu FTP wird die gesamte Sitzung verschlüsselt. Es werden keine Kennwörter in reiner Textform gesendet, wodurch SFTP viel weniger anfällig für Abhörangriffe durch Drittparteien ist.

Geben Sie einen optionalen SITE-Befehl an

Gibt an, dass SITE-Befehl verwendet wird, um Services aufzurufen, die dem Hostsystem eigen sind. Verwenden Sie dann das Feld "Standortparameter", um einen Ausdruck anzugeben, der Parameter für den SITE-Befehl zurückgibt.

Site-Parameter

Geben Sie einen CA Process Automation-Ausdruck an, der Parameter für den SITE-Befehl zurückgibt.

Ausgabeparameter

remoteFile

remoteHost

remoteUserId

remoteUserPassword

secureFtp

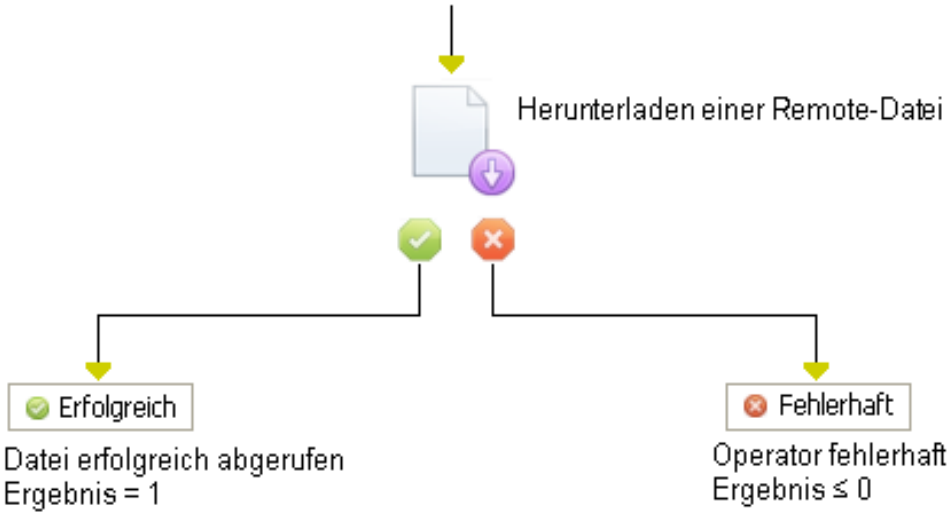
siteCommand

siteParameters

Operator "Datei herunterladen"

Der Operator "Datei herunterladen" kopiert eine Datei von einem Remote-Speicherort. Dieser Operator entspricht dem FTP-Befehl "get".

Der Remote-Host muss über einen konfigurierten FTP-Server verfügen. Zum Abrufen einer Remote-Datei müssen die angegebenen Anwender-Anmeldeinformationen über die entsprechenden Berechtigungen verfügen, um auf dem Remote-Host das Verzeichnis ändern und Dateien lesen zu können.



Eigenschaften von "Remote-Datei abrufen"

Lokale Datei

Gibt den Speicherort an, an dem die Datei lokal gespeichert werden soll. Beispiel: /IT PAM/import/script_ora1.sh.

Der Speicherort muss zur Laufzeit auf dem Kontaktpunkt, auf dem die Dateitransfer-Operatoren ausgeführt werden, gültig sein.

Wenn Sie die Windows-Dateinomenklatur verwenden, müssen vor umgekehrten Schrägstrichen folgendermaßen Maskierungszeichen eingefügt werden:

C:\\IT PAM\\import\\script_ora1.bat

Wir empfehlen den Einsatz von "standardisierten" Dateinamen, mit Schrägstrichen (/), auch beim Angeben eines Pfades auf einem Windows-Host. Beispiel: C:/IT PAM/import/script_ora1.bat.

Remote-Datei

Gibt den vollständigen Pfad für die Datei auf dem Remote-FTP-Host an. Beispiel: /tmp/IT PAM/scripts/script_ora1.sh.

Remote-Host

Gibt die IP-Adresse oder FTP-URL für den Remote-Host an. Für eine FTP-Site auf dem Intranet Ihres Unternehmens können Sie den Servernamen (//Servername) für die FTP-Site angeben.

Standardmäßig verwenden FTP-Sites Port 21. Sie können FTP-Sites auch private Ports zuweisen. Private Ports liegen zwischen 49152 und 65535. Um einen privaten FTP-Port anzugeben, fügen Sie am Ende der Adresse einen Doppelpunkt (:) und im Anschluss eine Nummer eines privaten Ports an. Auf die folgende Weise wird Port 50021 für einen Remote-FTP-Server angegeben:

```
172.24.36.107:50021
```

Remote-Anwender-ID

Gibt die Anwender-ID für den Zugriff auf den Remote-FTP-Host an.

Remote-Anwenderkennwort

Gibt das Kennwort an, um auf den Remote-FTP-Host zuzugreifen.

Binäre Übertragung

Gibt an, dass der FTP-Binärmodus für das Übertragen binärer Dateien verwendet werden soll. Zum Beispiel:

Wählen Sie ein Kontrollkästchen mit den folgenden Dateitypen aus:

- Ausführbare Dateien
- SPSS-Systemdateien
- SAS-Transport-Dateien
- Stata-Datensätze
- Grafikdateien

Von ASCII in EBCDIC konvertieren

Gibt an, ASCII-Zeichencode vor der Dateiübertragung in EBCDIC zu konvertieren. EBCDIC wird in einer z/OS-Umgebung verwendet, in der die Datei auf einem z/OS-Host gelesen werden kann.

Sicheres Dateiübertragungsverfahren (SFTP) verwenden

Öffnet eine sichere FTP (SFTP)-Sitzung. SFTP ähnelt FTP, aber im Gegensatz zu FTP wird die gesamte Sitzung verschlüsselt. Es werden keine Kennwörter in reiner Textform gesendet, wodurch SFTP viel weniger anfällig für Abhörangriffe durch Drittparteien ist.

CA Process Automation verwendet das SSH2/SFTP-Protokoll mit Anwendernamen/Kennwort-Authentifizierung. Das SSH2/SFTP-Protokoll unterstützt nur binäre Übertragungen.

Geben Sie einen optionalen SITE-Befehl an

Verwendet den SITE-Befehl, um Services aufzurufen, die dem Hostsystem eigen sind. Verwenden Sie dann das Feld "Standortparameter", um einen CA Process Automation-Ausdruck anzugeben, der Parameter für den SITE-Befehl zurückgibt. Diese Option wird beispielsweise verwendet, um Dateien auf einem Ziel-MVS-System zu bemessen.

Site-Parameter

Geben Sie einen CA Process Automation-Ausdruck an, der Parameter für den SITE-Befehl zurückgibt.

Ausgabeparameter

localFile

remoteFile

remoteHost

remoteUserId

remoteUserPassword

transferMode

secureFtp

siteCommand

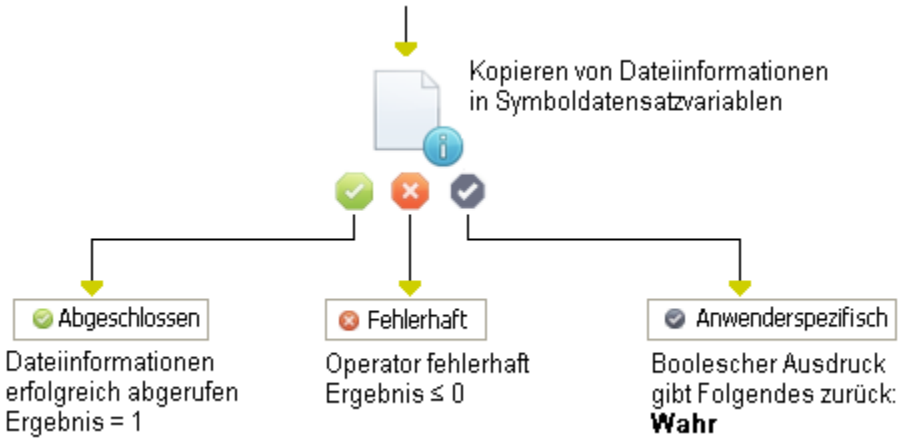
siteParameters

Operator "Dateiinformationen abrufen"

Der Operator "Dateiinformationen abrufen" ruft die Dateiattribute einer Remote-Datei ab und speichert sie in Variablen (z. B. Berechtigungen, Größe und Gruppe) in seinem Operatordatensatz.

Die Liste bedeutsamer Attribute ist vom Dateisystem abhängig. Um die Attribute für eine angegebene Datei oder einen Ordner anzuzeigen, zeigen Sie den Operatordatensatz an.

Der Remote-Host muss über einen konfigurierten FTP-Server verfügen. Die angegebenen Anwender-Anmeldeinformationen müssen über die entsprechenden Leseberechtigungen auf dem Remote-Host verfügen



Eingabeparameter

Remote-Datei

Gibt den Pfad für die Datei auf dem Remote-FTP-Host an. Beispiel: "/tmp/IT PAM/scripts/script_ora1.sh"

Remote-Host

Gibt die IP-Adresse oder FTP-URL für den Remote-Host an. Für eine FTP-Site auf dem Intranet Ihres Unternehmens können Sie den Servernamen (//Servername) für die FTP-Site angeben.

Standardmäßig verwenden FTP-Sites Port 21. Sie können FTP-Sites auch private Ports zuweisen. Private Ports liegen zwischen 49152 und 65535. Um einen privaten FTP-Port anzugeben, fügen Sie am Ende der Adresse einen Doppelpunkt (:) und im Anschluss eine Nummer eines privaten Ports an. Auf die folgende Weise wird Port 50021 für einen Remote-FTP-Server angegeben:

```
172.24.36.107:50021
```

Remote-Anwender-ID

Gibt eine Anwender-ID an, um auf den Remote-FTP-Host zuzugreifen.

Remote-Anwenderkennwort

Gibt das Kennwort an, um auf den Remote-FTP-Host zuzugreifen.

Sicheres Dateiübertragungsverfahren (SFTP) verwenden

Öffnet eine sichere FTP (SFTP)-Sitzung. SFTP ähnelt FTP, aber im Gegensatz zu FTP wird die gesamte Sitzung verschlüsselt. Es werden keine Kennwörter in reiner Textform gesendet, wodurch SFTP viel weniger anfällig für Abhörangriffe durch Drittparteien ist.

Geben Sie einen optionalen SITE-Befehl an

Gibt an, dass SITE-Befehl verwendet wird, um Services aufzurufen, die dem Hostsystem eigen sind. Verwenden Sie dann das Feld "Standortparameter", um einen CA Process Automation-Ausdruck anzugeben, der Parameter für den SITE-Befehl zurückgibt.

Site-Parameter

Geben Sie einen CA Process Automation-Ausdruck an, der Parameter für den SITE-Befehl zurückgibt.

Ausgabeparameter

remoteFile

remoteHost

remoteUserId

remoteUserPassword

secureFtp

siteCommand

siteParameters

FileExists

FileName

FileTime

IsDirectory

IsFile

IsSpecial

FileGroup

FileGroupExec

FileGroupRead

FileGroupWrite

FileOtherExec

FileOtherRead

FileOtherWrite

FileOwner

FileOwnerExec

FileOwnerRead

FileOwnerWrite

FilePermission

FileSize

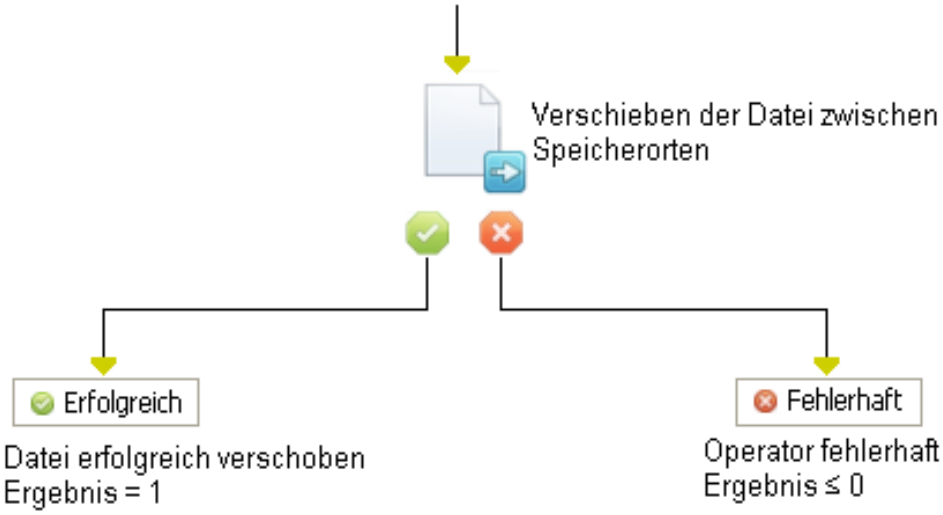
FileSizeKB

FileSizeMB

Operator "Datei verschieben"

Der Operator "Datei verschieben" verschiebt eine Datei von einem Remote-Speicherort auf einen anderen Remote-Speicherort auf dem gleichen Server. Sie können ihn verwenden, um eine Datei umzubenennen, indem Sie sowohl für den alten Namen als auch für den neuen Namen denselben Pfad angeben.

Der Remote-Host muss über einen konfigurierten FTP-Server verfügen. Zum Verschieben einer Datei müssen die angegebenen Anwender-Anmeldeinformationen über die entsprechenden Berechtigungen verfügen, um auf dem Remote-Host das Verzeichnis ändern und Dateien lesen zu können.



Eingabeparameter

Neuer Name

Gibt den Pfad und neuen Namen für die Datei auf dem Remote-FTP-Server an.

"/tmp/IT PAM/scripts/archived/IT PAM.new"

Aktueller Name

Gibt den vorhandenen Pfad und Namen für die Datei auf dem Remote-FTP-Server an. Beispiel: /tmp/IT-PAM/Skripten/IT PAM.old.

Remote-Host

Gibt die IP-Adresse oder FTP-URL für den Remote-Host an. Für eine FTP-Site auf dem Intranet Ihres Unternehmens können Sie den Servernamen (//Servername) für die FTP-Site angeben.

Standardmäßig verwenden FTP-Sites Port 21. Sie können FTP-Sites auch private Ports zuweisen. Private Ports liegen zwischen 49152 und 65535. Um einen privaten FTP-Port anzugeben, fügen Sie am Ende der Adresse einen Doppelpunkt (:) und im Anschluss eine Nummer eines privaten Ports an. Auf die folgende Weise wird Port 50021 für einen Remote-FTP-Server angegeben:

```
172.24.36.107:50021
```

Remote-Anwender-ID

Gibt eine Anwender-ID an, um auf den Remote-FTP-Host zuzugreifen.

Remote-Anwenderkennwort

Gibt das Kennwort an, um auf den Remote-FTP-Host zuzugreifen.

Sicheres Dateiübertragungsverfahren (SFTP) verwenden

Öffnet eine sichere FTP (SFTP)-Sitzung. SFTP ähnelt FTP, aber im Gegensatz zu FTP wird die gesamte Sitzung verschlüsselt. Es werden keine Kennwörter in reiner Textform gesendet, wodurch SFTP viel weniger anfällig für Abhörangriffe durch Drittparteien ist.

Geben Sie einen optionalen SITE-Befehl an

Gibt an, dass SITE-Befehl verwendet wird, um Services aufzurufen, die dem Hostsystem eigen sind. Verwenden Sie dann das Feld "Standortparameter", um einen CA Process Automation-Ausdruck anzugeben, der Parameter für den SITE-Befehl zurückgibt.

Site-Parameter

Geben Sie einen CA Process Automation-Ausdruck an, der Parameter für den SITE-Befehl zurückgibt.

Ausgabeparameter

newName

currentName

remoteHost

remoteUserId

remoteUserPassword

secureFtp

siteCommand

siteParameters

Operator "Datei über TFTP herunterladen"

Der Operator "Datei über TFTP herunterladen" empfängt über das TFTP-Protokoll eine Datei von einem Host auf einem Netzwerk . Der Host muss über einen ausgeführten TFTP-Server verfügen.

Es folgen Details zum TFTP-Protokoll:

- Verwendet UDP, um die Daten zu übertragen.
- Issues senden und warten auf ACKs.
- Führt Datentransfer normalerweise auf Port 69 aus.
- Sendet Daten normalerweise in einer Blockgröße von 512b oder kleiner.
- Verwendet keine Authentifizierung oder Autorisierung.

Wichtig! Testen Sie die TFTP-Funktionen außerhalb von CA Process Automation, bevor Sie sie in CA Process Automation-Prozessabläufe integrieren. Firewalls oder Routing kann UDP in Port 69 standardmäßig blockieren. Normalerweise ist der TFTP-Service entweder nicht installiert oder deaktiviert.

Eingabeparameter

Name des Remote-Host

Der Hostname oder die IP-Adresse des Remote-Hosts.

Remote-UDP-Port für TFTP

Der UDP-Port des TFTP-Services auf dem Remote-Host. Wenn dieser Parameter angegeben ist, überschreibt dieser Wert den Wert des Operatorkategoriefelds: Standardmäßiger UDP-Port für Trivial FTP. Wenn keine dieser Felder angegeben ist, verwendet der Operator den Standardwert 69.

Name der Remote-Datei

Der Name der Remote-Datei, die vom Remote-Host abgerufen werden soll. Dieser Parameter ist nicht der Pfad der Remote-Datei auf dem Remote-Host. Je nach Einrichtung ruft der TFTP-Server die Datei aus seinem Basisverzeichnis im Remote-Host ab.

Typ des Datentransfers

BIN für Binärdatei oder ASCII für ASCII (Text). Wenn dieser Typ nicht angegeben ist, verwendet der Operator den Standardwert: ASCII. Wenn ein anderer Wert angegeben ist (nicht ASCII oder BIN), verwendet der Operator den Standardwert "ASCII".

Lokale Datei für Download

Der vollständig qualifizierte Pfad der lokalen Datei (in dem Sie die Datei speichern, die über TFTP abgerufen wurde).

Lokale Portnummer (0 für anonymen Port)

Die lokale Portnummer, die verwendet werden soll, wenn Daten aus dem Remote-Host bzw. -Port abgerufen werden. Bei "0" wird ein anonymer Port verwendet. Wenn der Port angegeben und nicht verfügbar ist, kann der Betriebsablauf fehlschlagen. Wenn diese Zahl nicht angegeben ist, verwendet der Operator den Standardwert "0".

Zeitlimit (Sek.)

Der Wert für das Zeitlimit, das beim Eröffnen der Verbindung zum TFTP-Server verwendet werden soll. Wenn diese Zahl nicht angegeben ist, wird standardmäßig der Wert "20" verwendet.

Maximale Wiederholungen nach TFTP-Zeitüberschreitung

Die Anzahl, die der Betriebsablauf "Datei herunterladen" (nicht der gesamte CA Process Automation-Operator) nach der TFTP-Zeitüberschreitung maximal wiederholt werden soll. Wenn diese Zahl nicht angegeben ist, verwendet der Operator den Standardwert "5".

Ausgabeparameter

Ergebnis:

- 1: Wenn der Operator erfolgreich abgeschlossen wurde.
- -1: Wenn der Operator fehlgeschlagen ist.

Grund:

- Abgeschlossen: Wenn der Operator erfolgreich abgeschlossen wurde.
- Eine Fehlermeldung, wenn der Operator fehlgeschlagen ist.

remoteSSHHost

PORT

RemoteURL

TransferType

LocalFileName

LocalPort

Zeitlimit

MaxRetries

Operator-Ports

Erfolg

Die Ausführung des Operators wird erfolgreich abgeschlossen.

Fehlgeschlagen

Die Ausführung des Operators schlägt aus einem der folgenden Gründe fehl:

- Ungültiger Eingabeparameter vom Anwender. Das Feld "Grund" enthält eine Fehlermeldung mit dem Problem angibt.
- Die lokale Datei ist vorhanden, es kann aber nicht in die Datei geschrieben werden.
- Die lokale Datei ist ein Verzeichnis.
- Die lokale Datei ist nicht vorhanden. In das übergeordnete Verzeichnis der Datei kann nicht geschrieben werden, da die aktuellen Berechtigungen und Einschränkungen verhindern, dass CA Process Automation in das Verzeichnis schreibt.
- Die aktuellen Berechtigungen und Einschränkungen verhindern, dass CA Process Automation in die lokale Datei schreibt.
- Ein unbekannter Host wird angegeben.
- Beim Empfangen der Remote-Datei tritt ein E-/A-Fehler auf.

- Es tritt ein Zeitlimitfehler auf, wenn CA Process Automation keine Verbindung mit dem Remote-Host am angegebenen Remote-Port herstellen kann. In diesem Fall überschreitet der Operator das Zeitlimit nicht, da der TFTP-Client diesen Issue als E-/A-Fehler meldet (nicht als Zeitlimitfehler).
- Andere (im Feld "Grund" angegeben).

Anwenderspezifische Ports

Wenn während des Prozessdesigns durch den Anwender festgelegt.

Operator "Datei über TFTP hochladen"

Der Operator "Datei über TFTP hochladen" sendet über das TFTP-Protokoll eine Datei an einen Host auf einem Netzwerk . Der Host muss über einen ausgeführten TFTP-Server verfügen.

Es folgen Details zum TFTP-Protokoll:

- Verwendet UDP, um die Daten zu übertragen.
- Issues senden und warten auf ACKs.
- Führt Datentransfer normalerweise auf Port 69 aus.
- Sendet Daten normalerweise in einer Blockgröße von 512b oder kleiner.
- Verwendet keine Authentifizierung oder Autorisierung.

Wichtig! Testen Sie die TFTP-Funktionen außerhalb von CA Process Automation, bevor Sie sie in CA Process Automation-Prozessabläufe integrieren. Firewalls oder Routing kann UDP in Port 69 standardmäßig blockieren. Normalerweise ist der TFTP-Service entweder nicht installiert oder deaktiviert.

Eingabeparameter

Name des Remote-Host

Der Hostname oder die IP-Adresse des Remote-Hosts.

Remote-UDP-Port für TFTP

Der UDP-Port des TFTP-Services auf dem Remote-Host. Wenn dieser Port angegeben ist, überschreibt dieser Wert den Wert des Operatorkategoriefelds: Standardmäßiger UDP-Port für Trivial FTP. Wenn keine dieser Felder angegeben ist, verwendet der Operator den Standardwert 69.

Name der Remote-Datei

Der Name, der verwendet werden soll, wenn die Datei (die versendet wird) auf dem Remote-Host erstellt wird. Dieser Parameter ist nicht der Pfad der Remote-Datei auf dem Remote-Host. Je nach Einrichtung speichert der TFTP-Server die Datei in das Basisverzeichnis im Remote-Host.

Typ des Datentransfers

BIN für Binärdatei oder ASCII für ASCII (Text). Wenn nicht angegeben, verwendet der Operator den Standardwert: ASCII. Wenn ein anderer Wert angegeben ist (nicht ASCII oder BIN), verwendet der Operator den Standardwert "ASCII".

Lokale Datei für Upload

Der vollständig qualifizierte Pfad der lokalen Datei, die über TFTP gesendet werden soll.

Lokale Portnummer (0 für anonymen Port)

Die lokale Portnummer, die verwendet werden soll, wenn Daten an den Remote-Host bzw. -Port gesendet werden. Bei "0" wird ein anonymer Port verwendet. Wenn der Port angegeben und nicht verfügbar ist, kann der Betriebsablauf fehlschlagen. Wenn diese Zahl nicht angegeben ist, verwendet der Operator den Standardwert "0".

Zeitlimit (Sek.)

Der Wert für das Zeitlimit, das beim Eröffnen der Verbindung zum TFTP-Server verwendet werden soll. Wenn diese Zahl nicht angegeben ist, wird standardmäßig der Wert "20" verwendet.

Maximale Wiederholungen nach TFTP-Zeitüberschreitung

Die Anzahl, die der Betriebsablauf "Datei laden" (nicht der gesamte CA Process Automation-Operator) nach der TFTP-Zeitüberschreitung maximal wiederholt werden soll. Wenn diese Zahl nicht angegeben ist, verwendet der Operator den Standardwert "5".

Ausgabeparameter

Ergebnis:

- 1: Wenn der Operator erfolgreich abgeschlossen wurde.
- -1: Wenn der Operator fehlgeschlagen ist.

Grund:

- Abgeschlossen: Wenn der Operator erfolgreich abgeschlossen wurde.
- Eine Fehlermeldung, wenn der Operator fehlgeschlagen ist.

remoteSSHHost

PORT

RemoteURL

TransferType

LocalFileName

LocalPort

Zeitlimit

MaxRetries

Operator-Ports

Erfolg

Die Ausführung des Operators wird erfolgreich abgeschlossen.

Fehlgeschlagen

Die Ausführung des Operators schlägt aus einem der folgenden Gründe fehl:

- Ungültiger Eingabeparameter vom Anwender. Das Feld "Grund" enthält eine Fehlermeldung mit dem Problem angibt.
- Die lokale Datei existiert nicht, ist ungültig oder kann nicht gelesen werden.
- Die aktuellen Berechtigungen und Einschränkungen verhindern, dass CA Process Automation die lokale Datei liest.
- Beim Senden der lokalen Datei tritt ein E-/A-Fehler auf.
- Ein unbekannter Host wird angegeben.
- Es tritt ein Zeitlimitfehler auf, wenn CA Process Automation keine Verbindung mit dem Remote-Host am angegebenen Remote-Port herstellen kann. In diesem Fall überschreitet der Operator das Zeitlimit nicht, da der TFTP-Client diesen Issue als E-/A-Fehler meldet (nicht als Zeitlimitfehler).
- Andere (im Feld "Grund" angegeben).

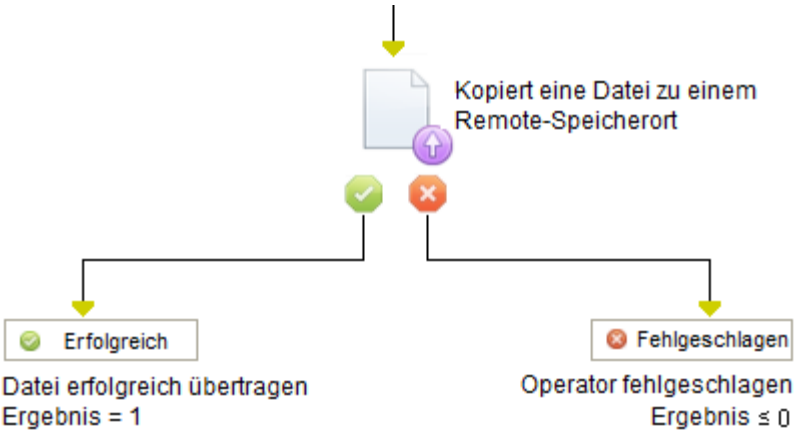
Anwenderspezifische Ports

Wenn während des Prozessdesigns durch den Anwender festgelegt.

Operator "Datei laden"

Der Operator "Datei laden" kopiert eine Datei von einem lokalen Speicherort auf einen Remote-Computer. Diese Aktion entspricht dem FTP-Befehl "put".

Damit ein Verzeichnis hochgeladen werden kann, müssen die angegebenen Anwenderanmeldeinformationen über die entsprechenden Berechtigungen zum Ändern des Verzeichnisses und Schreiben auf dem Remote-Host verfügen. Der Remote-Host muss über einen konfigurierten FTP-Server verfügen.



Eingabeparameter

Lokale Datei

Geben Sie den vollständigen Pfad der zu übertragenden lokalen Datei an. Beispiel: C:\\IT PAM\\import\\script_ora1.sh. Diese Option ist nicht verfügbar, wenn Sie das Kontrollkästchen "Inline-Daten" auswählen.

Inline-Daten

Wenn diese Option ausgewählt ist, wird mit dem Operator in der Bibliothek gespeicherte Text übertragen. Klicken Sie auf das Feld "Inline-Daten", um Daten einzugeben.

Inline-Daten

Klicken Sie auf dieses Feld, um den Inline-Daten-Editor zu öffnen, wo Sie die zu übertragenden Daten eingeben können.

Remote-Datei

Gibt den Pfad für die Datei auf dem Remote-FTP-Host an. Beispiel: /tmp/IT PAM/scripts/script_ora1.sh. Wenn Sie keinen Dateinamen angeben, speichert der Operator die Datei unter dem gleichen Namen wie die kopierte Datei. Der Speicherort muss zur Laufzeit auf dem Kontaktpunkt, auf dem die Dateitransfer-Operatoren ausgeführt werden, gültig sein.

Remote-Host

Gibt die IP-Adresse oder FTP-URL für den Remote-Host an. Für eine FTP-Site auf dem Intranet Ihres Unternehmens können Sie den Servernamen (//Servername) für die FTP-Site angeben.

Standardmäßig verwenden FTP-Sites Port 21. Sie können FTP-Sites auch private Ports zuweisen. Private Ports liegen zwischen 49152 und 65535. Um einen privaten FTP-Port anzugeben, fügen Sie am Ende der Adresse einen Doppelpunkt (:) und im Anschluss eine Nummer eines privaten Ports an. Auf die folgende Weise wird Port 50021 für einen Remote-FTP-Server angegeben:

```
172.24.36.107:50021
```

Remote-Anwender-ID

Gibt eine Anwender-ID an, um auf den Remote-FTP-Host zuzugreifen.

Remote-Anwenderkennwort

Gibt das Kennwort an, um auf den Remote-FTP-Host zuzugreifen.

Binäre Übertragung

Verwendet den FTP-Binärmodus für das Übertragen binärer Dateien. Wenn diese Option ausgewählt ist, werden beispielsweise die folgenden Dateitypen verwendet:

- Ausführbare Dateien
- SPSS-Systemdateien
- SAS-Transport-Dateien
- Stata-Datensätze
- Grafikdateien

Sicheres Dateiübertragungsverfahren (SFTP) verwenden

Wählen Sie dieses Kontrollkästchen aus, um eine Sitzung mit sicherem Dateiübertragungsverfahren (SFTP) zu öffnen. SFTP ähnelt FTP, aber im Gegensatz zu FTP wird die gesamte Sitzung verschlüsselt. Es werden keine Kennwörter im Klartextformat gesendet, wodurch SFTP viel weniger anfällig für Abhörangriffe durch Drittparteien ist.

Geben Sie einen optionalen SITE-Befehl an

Wählen Sie dieses Kontrollkästchen aus, um den SITE-Befehl zu verwenden, um hostsystemspezifische Services aufzurufen. Verwenden Sie dann das Feld "Standortparameter", um einen CA Process Automation-Ausdruck anzugeben, der Parameter für den SITE-Befehl zurückgibt.

Site-Parameter

Geben Sie einen CA Process Automation-Ausdruck an, der Parameter für den SITE-Befehl zurückgibt.

Ausgabeparameter

localFile

isInline

Inline Data

remoteFile

remoteHost

remoteUserId

remoteUserPassword

transferMode

secureFtp

siteCommand

siteParameters

Kapitel 11: Java-Verwaltung

Die Java-Verwaltungs-Operatoren bieten eine Verwaltungsschnittstelle für externe Systeme, die JMX unterstützen.

JMX-Anmeldeparameter

Die JMX-Anmeldeparameter werden für jeden der Java-Management-Operatoren angezeigt. Diese Parameter konfigurieren die für die Anmeldung und die Kommunikation mit dem JMX-Server erforderlichen Einstellungen.

Vom Anwender erstellte JMX-Service-URL verwenden

Aktivieren Sie dieses Kontrollkästchen, um eine JMX-Service-URL anstatt eines Servers anzugeben.

Wenn Sie dieses Kontrollkästchen aktivieren, werden die folgenden Felder aktiviert:

- JMX-Service-URL
- Remote-JMX-Anwender
- Remote-JMX-Kennwort

Wenn Sie dieses Kontrollkästchen aktivieren, werden die folgenden Felder deaktiviert:

- Remote-JMX-Host
- Remote-Registrierungsport
- Remote-JMX-Server

JMX-Service-URL

Gibt die URL für einen JMX-Service an.

URL-Muster:
service:jmx:rmi:///jndi/rmi://<*ZIELCOMPUTER*>:<*RMI_REGISTRIERUNGSPORT*>/JMX RMIServer

Beispiel:

ZIELCOMPUTER: PA-w2k3-3

RMI_REGISTRIERUNGSPORT: 9999

JMX RMI Server: Server

Die URL sieht wie folgt aus:

service:jmx:rmi:///jndi/rmi://PA-w2k3-2:9999/server

Remote-JMX-Host

Gibt den Hostcomputernamen für den JMX-Agenten an. Sie können den vollständigen Computernamen oder eine IP-Adresse auf Ihrem Netzwerk angeben.

Wenn der JMX-Agent auf dem gleichen Computer wie CA Process Automation ausgeführt wird, ist "localhost" der Standard.

RMI-Registrierungsport

Gibt den Verbindungsport für den JMX-Agenten an.

Der Standardport ist 12345. Sie können jedoch den JMX-Agenten auf einem von Ihnen definierten RMI-Registrierungsport ausführen.

Remote-JMX-Server

Gibt den Namen des JMX-Servers an.

Remote-JMX-Anwender

Gibt den Anwendernamen zum Herstellen einer Verbindung mit dem JMX-Agenten auf dem MBean-Server an. Wenn für den JMX-Agenten Sicherheit deaktiviert wird, lassen Sie diese Option leer.

Remote-JMX-Kennwort

Gibt das Kennwort an, das verwendet wird, um mit dem JMX-Agenten auf dem MBean-Server eine Verbindung herzustellen. Wenn für den JMX-Agenten Sicherheit deaktiviert wird, lassen Sie diese Option leer.

Operator "MBean-Attribute abrufen"

Der Operator "MBean-Attribute abrufen" ruft den Wert eines MBean über JMX von einem Knoten ab. Der Operator "Vorlage" benötigt den Namen des MBean und Kontaktinformationen und Anmeldeinformationen für den JMX-Server. Die abgerufenen Werte werden in den Prozessdatensatz eingefügt. Der Operator-Datensatzvariable wird der Name des Attributs als Name und der abgerufene Wert als Wert zugewiesen.

Eingabeparameter

Managed Bean-Name

Gibt den Namen des Management Bean an, auf das zugegriffen werden soll.

Wählen Sie in der Drop-down-Liste den Namen des Management Bean aus, auf das zugegriffen werden soll, oder geben Sie den anwenderspezifischen MBean-Namen manuell ein.

Managed Bean-Attribut

Wählen Sie den Namen des abzurufenden Attributs in der Liste aus, oder geben Sie ihn ein.

JMX-Anmeldeparameter (siehe Seite 349)

Ausgabeparameter

ManagedBeanName

ManagedBeanAttribute

State

UserSpecifiedURL

RemoteJMXURL

RemoteHost

RemoteRMIRegistryPort

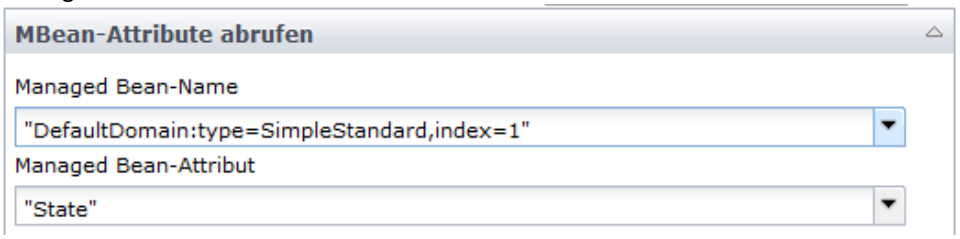
RemoteJMXServer

RemoteUser

RemotePassword

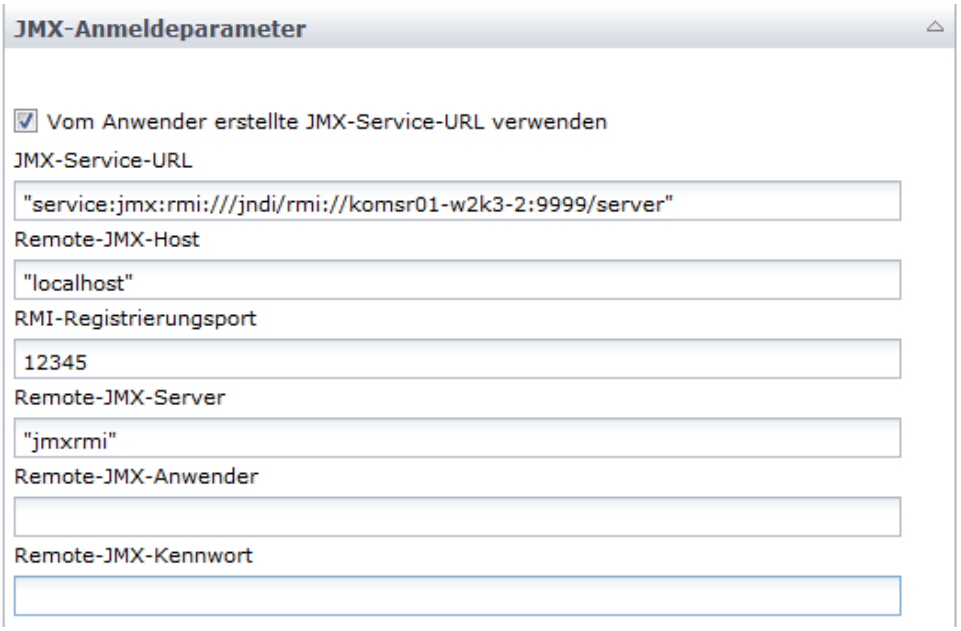
Beispiel

Dieses Beispiel zeigt einen anwenderspezifischen Managed Bean-Namen und ein Managed Bean-Attribut.

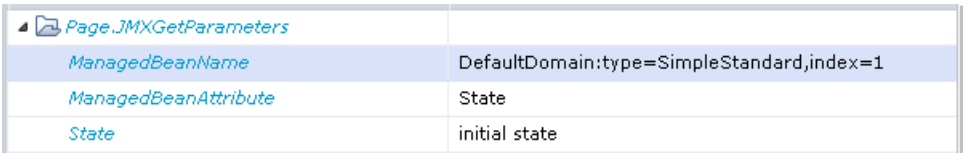


In den JMX-Anmeldeparametern wird eine Remote-JMX-Service-URL angegeben. Die Felder "Remote-JMX-Host", "RMI-Registrierungsport" und "Remote-JMX-Server" sind als ein Ergebnis standardmäßig deaktiviert.

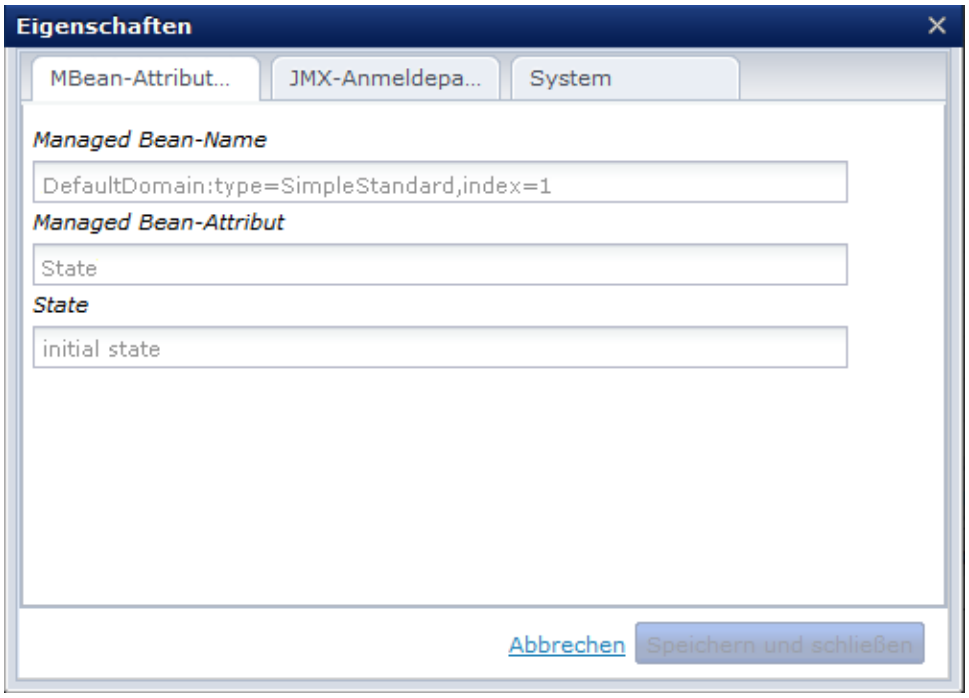
Die Felder "Remote-JMX-Anwender" und "Remote-JMX-Kennwort" sind leer, da der Server keine Sicherheit aktiviert hat.



Nachdem der Operator erfolgreich ausgeführt wurde, gibt der Zustands-Parameter einen Anfangsstatus von dem MBean zurück, wie im Ausgabedatensatz des Operators im Datensatzbereich gezeigt wird:



Diese Informationen werden auch in den Ausgabevariableneigenschaften des Operators im Bereich "Protokoll" angezeigt:



Operator "MBean-Methode aufrufen"

Der Operator "MBean-Methode aufrufen" ruft eine Methode auf einem JMX-Server auf.

Eingabeparameter

Managed Bean-Name

Gibt den Namen des Management Bean an, auf das zugegriffen werden soll.

Wählen Sie in der Drop-down-Liste den Namen des Management Bean aus, auf das zugegriffen werden soll, oder geben Sie den anwenderspezifischen MBean-Namen manuell ein.

Managed Bean-Methode

Gibt die Methode an, die aufgerufen werden soll.

Methodenparameter

Gibt Parameter für die Methode an.

Verwenden Sie die Schaltflächen in diesem Feld, um Parameter hinzuzufügen, zu entfernen oder umzuordnen.

JMX-Anmeldeparameter (siehe Seite 349)

Ausgabeparameter

ManagedBeanName

ManagedBeanMethod

ManagedBeanMethodParams

InvokeResults

UserSpecifiedURL

RemoteJMXURL

RemoteHost

RemoteRMIRegistryPort

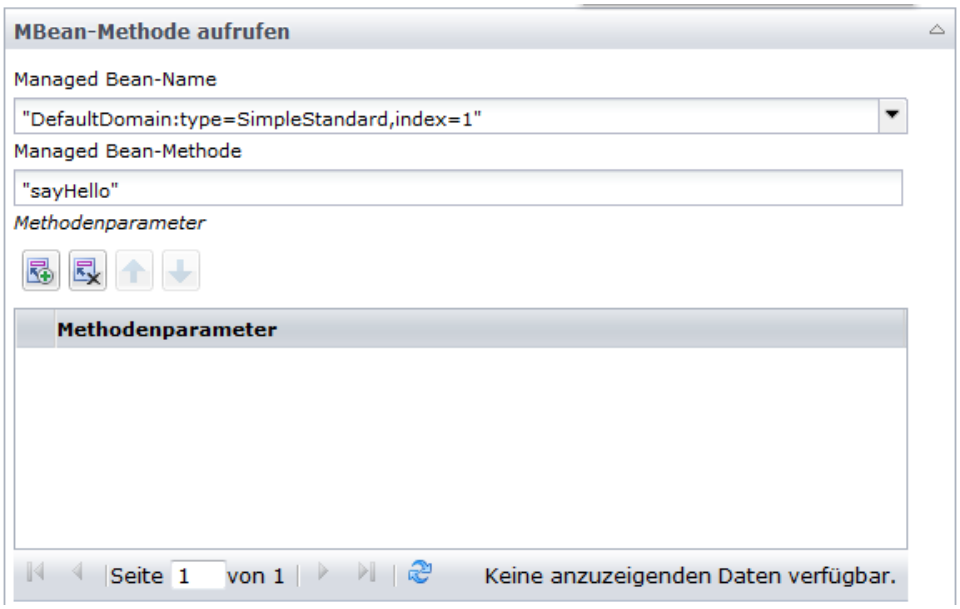
RemoteJMXServer

RemoteUser

RemotePassword

Beispiel

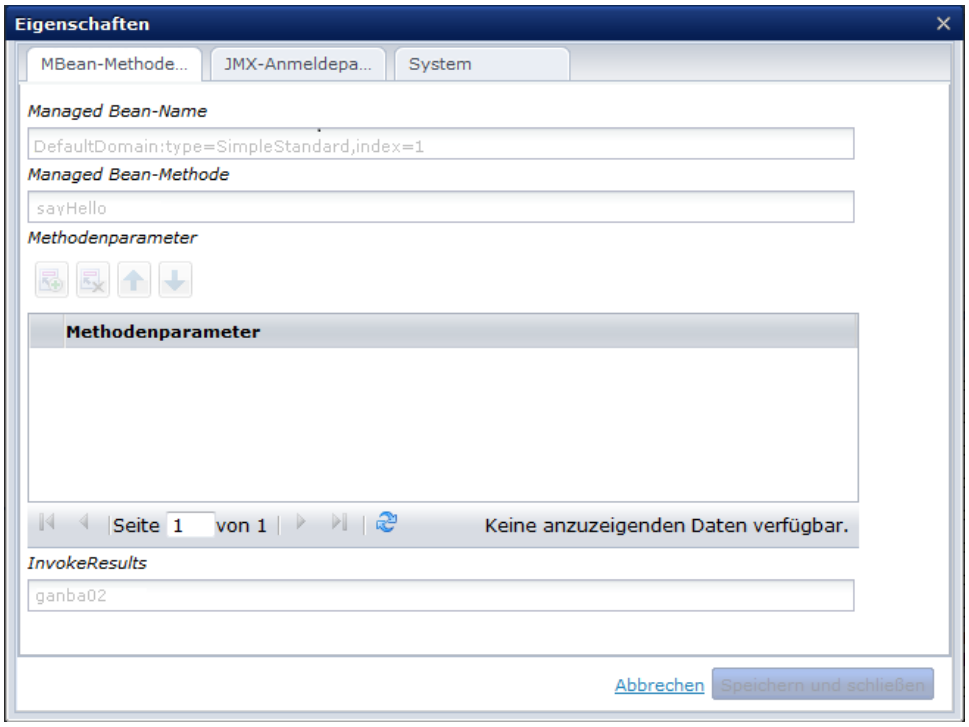
In diesem Beispiel wird die MBean-Methode "sayHello" ohne Parameter vom anwenderspezifischen MBean aufgerufen.



Nachdem der Operator erfolgreich ausgeführt wurde, gibt die MBean-Methode den Wert zum InvokeResults-Parameter zurück, wie im Ausgabedatensatz des Operators im Datensatzbereich gezeigt wird:

Page.JMXInvokeParameters	
ManagedBeanName	DefaultDomain:type=SimpleStandard,index=1
ManagedBeanMethod	sayHello
ManagedBeanMethodParams	[0]
InvokeResults	ganba02

Diese Informationen werden auch in den Ausgabevariableneigenschaften des Operators im Bereich "Protokoll" angezeigt:



Operator "MBean-Attribute aktualisieren"

Der Operator "MBean-Attribute aktualisieren" legt den MBean-Attributwert auf das MBean-Attribut fest.

Eingabeparameter

Managed Bean-Name

Gibt den Namen des Management Bean an, auf das zugegriffen werden soll.

Wählen Sie in der Drop-down-Liste den Namen des Management Bean aus, auf das zugegriffen werden soll, oder geben Sie den anwenderspezifischen MBean-Namen manuell ein.

Managed Bean-Attribut

Gibt den Namen eines zu aktualisierenden JMX-MBean-Attributs an.

Attributwert

Gibt einen Wert an, der als Wert des JMX-Attributs festgelegt wird.

JMX-Anmeldeparameter (siehe Seite 349)

Ausgabeparameter

ManagedBeanName

ManagedBeanAttribute

ManagedBeanAttributeValue

UserSpecifiedURL

RemoteJMXURL

RemoteHost

RemoteRMIRegistryPort

RemoteJMXServer

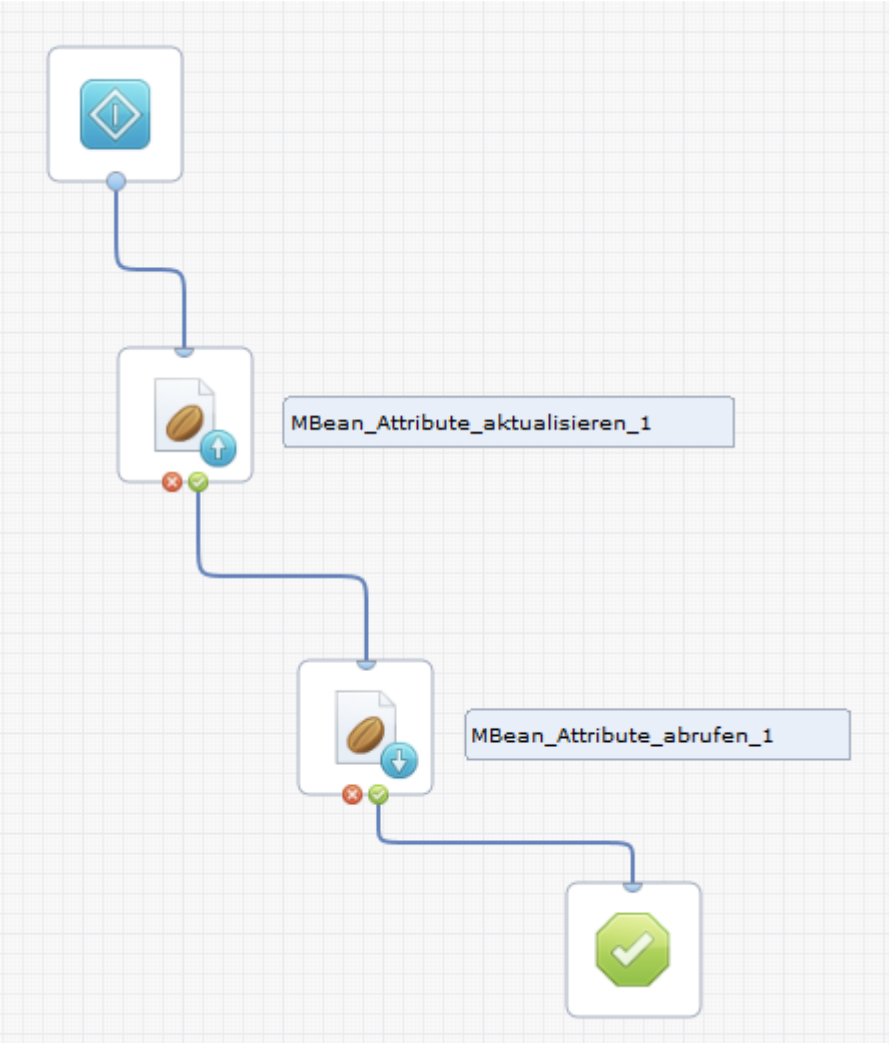
RemoteUser

RemotePassword

Beispiel

In diesem Beispiel wird Folgendes veranschaulicht:

- Das Aktualisieren des MBean-Attributwerts von "initial" zu "intermediate" mit dem Operator "MBean-Attribute aktualisieren".
- Das Empfangen des aktualisierten Werts des MBean-Attributzustands mit dem Operator "MBean-Attribute abrufen".

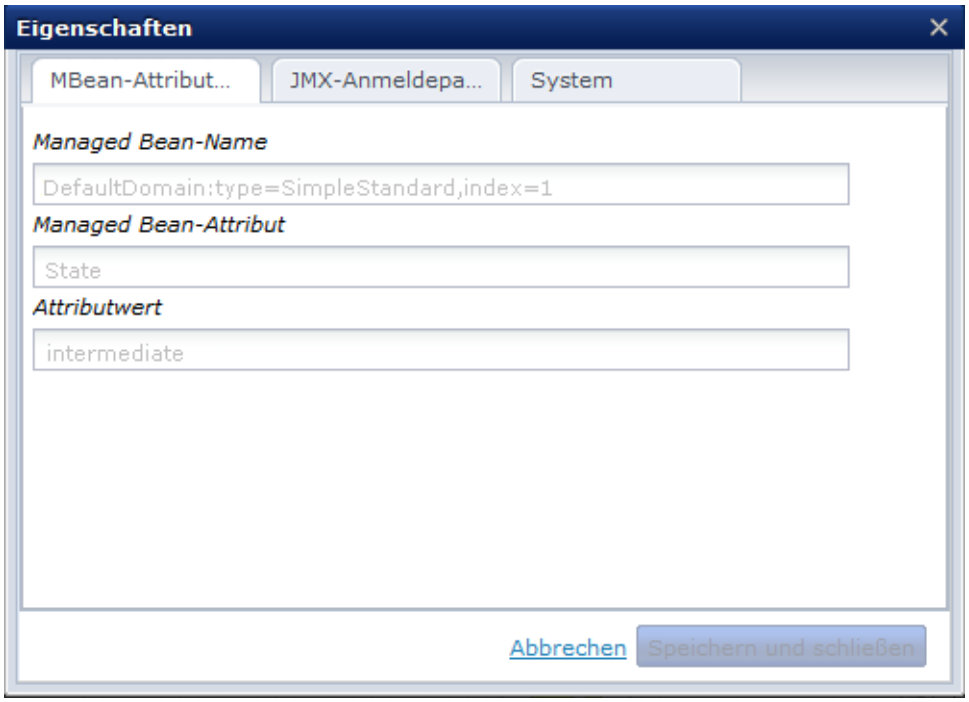


Diese Parameter sind die Eingabeparameter für den Operator "MBean-Attribute aktualisieren":

Nachdem der Operator erfolgreich ausgeführt wurde, wird der Wert des MBean-Attributzustands auf "intermediate" gesetzt, wie im Ausgabedatensatz des Operators im Datensatzbereich gezeigt wird:

Page.JMXSetParameters	
ManagedBeanName	DefaultDomain:type=SimpleStandard,index=1
ManagedBeanAttribute	State
ManagedBeanAttributeValue	intermediate

Diese Informationen werden auch in den Ausgabevariableneigenschaften des Operators im Bereich "Protokoll" angezeigt:



Diese Variablen sind die Operatorausgabe-Datensatzvariablen vom Datensatzbereich des Operators "MBean-Attribute abrufen". Der "State"-Wert wurde auf "intermediate" aktualisiert.

Page.JMXGetParameters	
ManagedBeanName	DefaultDomain:type=SimpleStandard,index=1
ManagedBeanAttribute	State
State	intermediate

Kapitel 12: Netzwerkhilfsprogramme

Die Netzwerkhilfsprogramm-Operatoren stellen verschiedene standardmäßige Netzwerkprotokoll-Hilfsprogramme für automatisierte Business-Prozesse bereit, die von CA Process Automation ermöglicht werden. Diese Operatoren stellen allgemeine Hilfsprogramme bereit, die verschiedene Netzwerkschnittstellen und Geräte validieren. Diese Operatoren können auch für Remote-Services und Server ausgeführt werden und verschieben Daten als Teil des allgemeinen Automatisierungsprozesses, wodurch manuelle Validierung und Verifizierung reduziert werden.

Die Netzwerkhilfsprogramm-Operatoren haben die Möglichkeit, anwenderspezifische Ausgangsports zu erstellen. Sie können anwenderspezifische Ports (nicht automatische Ausgangsports) auf dem Operator festlegen, wenn Sie den Prozess erstellen. Diese Ports sind zusätzliche Ports neben den automatischen Ausgangsports, die alle Operatoren haben (erfolgreiche und fehlgeschlagene Ports).

Verwenden Sie die Operatoren "Netzwerkhilfsprogramme", um systemeigene Hilfsprogramme für Netzwerkschnittstellen zu verwenden (statt hostbasierten Skripterstellungssprachen und anderen hostbasierten Hilfsprogrammen).

Operator "Lokale Netzwerkschnittstellen abrufen"

Der Operator "Lokale Netzwerkschnittstellen abrufen" listet alle Netzwerkschnittstellen im lokalen Host auf. Der lokale Host ist der Host, auf dem der Kontaktpunkt für den Operator ausgeführt wird.

Für jede Netzwerkschnittstelle listet der Operator Folgendes auf:

- Name der Schnittstelle
- MAC-Adresse
- Anzeigename
- Liste der InetAddresses, die der Schnittstelle zugeordnet sind.

  Hinweis: Jede InetAddress besteht aus seinem kanonischen Namen, dem Hostnamen und der IP-Adresse.

Eingabeparameter

Es ist keine Operatoreingabe erforderlich.

Ausgabeparameter

StartTime

StartDate

NetworkInterfaces

Gibt ein Array mit ValueMaps an, das die Netzwerkschnittstellen enthält, die dem lokalen Host zugeordnet sind.

Jede Wertzuordnung in "NetworkInterfaces" enthält Folgendes:

Name

Gibt den Namen der Netzwerkschnittstelle an.

Anzeige

Gibt den Anzeigenamen der Netzwerkschnittstelle an.

MacAddress

Gibt die Hardwareadresse, üblicherweise MAC-Adresse, der Netzwerkschnittstelle an. Ein leeres Feld zeigt an, dass der Operator die MAC-Adresse aus der Netzwerkschnittstelle nicht abrufen kann. Das Abrufen kann fehlschlagen, wenn Berechtigungen oder eine MAC-Adresse für diese Netzwerkschnittstelle fehlen.

InetAddresses

Gibt ein Array mit ValueMaps an, das die InetAddresses enthält, die der Netzwerkschnittstelle zugeordnet ist. Jede Wertzuordnung in "InetAddresses" enthält Folgendes:

IpAddress

Gibt die IP-Adresse an.

Host

Gibt den Hostnamen an. Die IP-Adresse wird zurückgegeben, wenn der Operator aufgrund des Netzwerk-Setups keine umgekehrte Suche ausführen kann.

CanonicalName

Gibt den kanonischen Namen an. Wenn der Operator den FQDN nicht abrufen kann, wird eine IP-Adresse zurückgegeben.

Ergebnis

Eines der folgenden Elemente:

1

Zeigt an, dass der Operator erfolgreich abgeschlossen wurde.

-1

Zeigt an, dass der Operator fehlschlagen ist.

Ursache

Eines der folgenden Elemente:

Abgeschlossen

Zeigt an, dass der Operator erfolgreich abgeschlossen wurde.

<error message>

Gibt in einer Fehlermeldung an, warum der Operator fehlgeschlagen ist.

Operator-Ports

Erfolg

Der Operator wurde erfolgreich abgeschlossen.

Fehlgeschlagen

Der Operator schlägt aus einem der folgenden Gründe fehl:

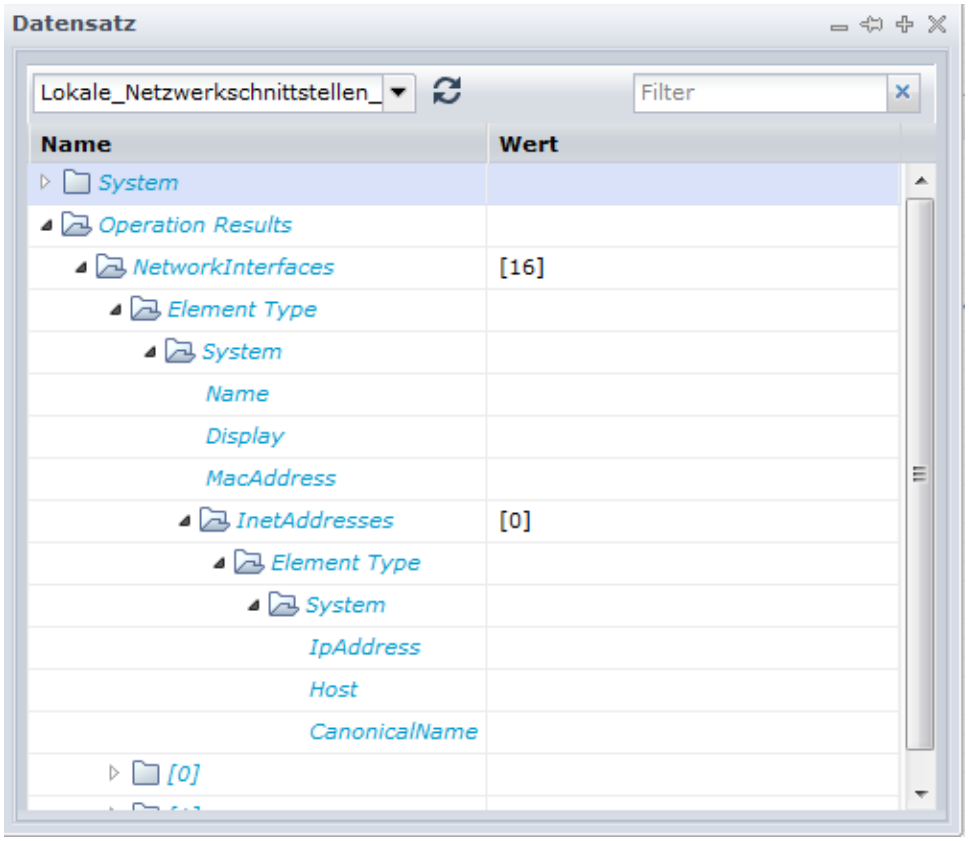
- Die Liste von lokalen Netzwerkschnittstellen des lokalen Hosts kann nicht abgerufen werden.
- Andere Gründe, die in Fehlermeldungen angegeben sind.

Anwenderspezifische Ports

Wenn während des Prozessdesigns durch den Anwender festgelegt.

Beispiel

Beispiel-ValueMap: "Network Interfaces" und "InetAddresses"



Operator "Status des Netzwerkdienstes abrufen"

Der Operator "Status des Netzwerkdienstes abrufen" ermöglicht es Ihnen, über TCP oder UDP mit einem lokalen oder Remote-Service zu kommunizieren. Sie können diesen Operator verwenden, um Daten zu senden und anschließen eine Antwort zu erhalten. Dieser Operator kann die Antwort gegen ein vordefiniertes Muster validieren, um zu bestimmen, ob der Netzwerkdienst abgelaufen ist. Dieser Validierungstyp ermöglicht es dem Operator, über den Status des Services und den Computer zu berichten, der diesen Service hostet.

Hinweis: Sie können mithilfe dieses Operators keine binäre Meldung senden, da nur Zeichenfolgen (Text) unterstützt werden.

TCP ist ein verbindungsorientiertes Protokoll. Mit dem Operator können Sie eine Verbindung mit dem Service herstellen, Daten senden und anschließend eine Antwort erhalten, die mit einem Muster abgeglichen werden kann. Ein anderer Service kann den gleichen Port abhören. Deswegen bedeutet eine erfolgreiche Verbindung nicht unbedingt, dass ein bestimmter Service ausgeführt wird.

UDP ist ein verbindungsloses Protokoll. Der Operator stellt keine Verbindung mit dem Service her, um den Status über UDP abzurufen. Um den Status abzurufen, sendet der Operator eine UDP-Meldung, liest die Antwort, und stellt sicher, dass die Antwortdaten mit einem Muster übereinstimmen.

Eingabeparameter

Name des Remote-Host

Gibt den Hostnamen oder FQDN des Computers an, der den Service hostet.

Standard

Leer - Zeigt an, dass der Operator annimmt, dass der Service auf dem Kontaktpunkthost ausgeführt wird.

Remote-Port

Gibt den Host-Port an, den der Service abhört. Einige bekannte und registrierte Ports sind Folgende:

- 21: FTP - File Transfer
- 22: SSH - The Secure Shell (SSH) Protocol
- 23: Telnet
- 2483: Oracle TTC
- 25: SMTP - Simple Mail Transfer
- 3306: MySQL
- 69: TFTP - Trivial File Transfer
- 80: HTTP
- 1433: Microsoft SQL Server

Sie können einen beliebigen gültigen Port in diesem Feld angeben.

Lokale Portnummer (0 für anonymen Port)

Gibt den lokalen Port an, den der Operator "Status des Netzwerkdienstes abrufen" auf dem Kontaktpunkthost verwendet, um eine Verbindung mit dem Remote-Port herzustellen.

Werte

- 0 - Zeigt an, dass ein anonymer Port verwendet wird.
- Leer - Entspricht dem Wert 0.

Hinweis: Wenn der angegebene Port nicht verfügbar ist, schlägt der Operator fehl.

Zu verwendendes Protokoll

Gibt das zu verwendende Protokoll an, wenn der Status des Service überprüft und Daten gesendet werden. Wenn UDP angegeben ist, wird das Feld "Zeitlimit für Verbindung (Sek.)" deaktiviert, da UDP ein verbindungsloses Protokoll ist.

Werte

Dieser Wert kann Folgendes sein:

- TCP
- UDP

Standard

TCP (wenn dieses Feld leer gelassen wird).

Zeitlimit für Verbindung (Sek.)

Gibt an, wie lange der Operator maximal auf eine Verbindung mit dem Service warten soll, bevor der Operator das Zeitlimit überschreitet. Dieses Feld gilt nur für TCP-Protokoll.

Werte

Dieser Wert kann Folgendes sein:

- 0 (null) - Zeigt kein Zeitlimit an.
- Eine beliebige positive Ganzzahl (in Sekunden)

Standard

20 Sekunden (wenn leer)

Zu sendende Daten

(Optional) Gibt Daten an, die dem Service gesendet werden sollen. Die meisten Services erwarten keine Daten.

Daten von Service lesen?

Gibt an, ob der Operator Daten aus dem Service liest, nachdem eine Verbindung hergestellt wurde.

Werte

Dieser Wert kann Folgendes sein:

- Ausgewählt - Zeigt an, dass der Operator Daten aus dem Service liest, nachdem eine Verbindung hergestellt wurde. Wählen Sie diese Option aus, um die Felder "Zeit bis zum Lesen von Daten (Sek.)", "Daten bis zum Lesen maximal (Byte)" und "Übereinzustimmendes Antwortmuster" zu aktivieren.
- Deaktiviert - Zeigt an, dass der Operator keine Daten aus dem Service liest, nachdem eine Verbindung hergestellt wurde. Dieser Wert ist für Services geeignet, die keine Daten zurückgeben.

Hinweise:

- Mit UDP werden alle Daten (wenn vorhanden) gleichzeitig gelesen. Wenn keine Daten vorhanden sind, wartet der Operator bis die "Zeit bis zum Lesen von Daten" abgelaufen ist. "Daten bis zum Lesen maximal" wird gleichzeitig in einer Zeichenfolge in der Länge "Daten bis zum Lesen maximal" gelesen.
- Mit TCP liest der Operator die Daten in Blöcken, bis einer der folgenden Grenzwerte erreicht wird:
  - Zeit bis zum Lesen von Daten
  - Daten bis zum Lesen maximal

Zeit bis zum Lesen von Daten (Sek.)

Gibt die Zeit an, die während der auf Antwortdaten vom Service gewartet wird. Dieses Feld wird angegeben, weil die Daten vom Service am Ende kein EOF (End of File) haben.

Werte

Dieser Wert ist eine positive Ganzzahl. Null (0) ist *nicht* zulässig. Anderenfalls würde der Operator sehr lange warten, bis der Service das Socket schließt.

Standard

20 Sekunden (wenn leer).

Daten bis zum Lesen maximal (Byte)

Gibt die Datenmenge an, die maximal vom Service gelesen wird.

Standard

4096 Byte

Übereinzustimmendes Antwortmuster

(Optional) Gibt ein Muster an, das zur Übereinstimmung der vom Service zurückgegebenen Daten verwendet werden soll. Wenn diese Option angegeben ist, bestimmt die Musterübereinstimmung, ob der Operator erfolgreich ist oder fehlschlägt.

Die Operator stimmt das Muster mit den Daten überein, die während der angegebenen Zeit in "Zeit bis zum Lesen von Daten" bis zur Anzahl der angegebenen Byte in "Daten bis zum Lesen maximal" gelesen werden.

Der Operator entspricht dem Muster als untergeordnete Zeichenfolge der Antwortdaten.

- Um mit Antwortdaten übereinzustimmen, die mit einer bestimmten Sequenz beginnen, verwenden Sie am Anfang des Musters ^.
- Um mit Antwortdaten übereinzustimmen, die mit einer bestimmten Sequenz enden, verwenden Sie am Ende des Musters $.
- Um eine neuen Zeilenabschluss abzugleichen, verwenden Sie einen Punkt (.). Der Punkt wird verwendet, um mehrzeilige Antwortdaten abzustimmen.

Ausgabeparameter

ReplyMessage

Enthält die vom Service empfangenen Daten. Dieses Feld ist leer, wenn der Service keine Daten gesendet hat oder wenn der Operator keine Daten aus dem Service liest.

LocalAddressInfo

Information bezüglich der lokalen Adresse, die verwendet wird, um folgendermaßen eine Verbindung mit dem Service herzustellen:

```
localhost/IP:port
```

Mit diesen Informationen können Sie die tatsächliche lokale Portnummer identifizieren, die verwendet wird, wenn die Lokale Portnummer (siehe Seite 365) auf 0 (für anonym) festgelegt wird.

RemoteAddressInfo

Informationen bezüglich der Remote-Adresse, mit der der Operator folgendermaßen verbunden ist:

```
Hostname/IP:port
```

Hinweis: UDP ist ein verbindungsloses Protokoll. Dieses Feld ist leer, wenn UDP in Zu verwendendes Protokoll (siehe Seite 365) ausgewählt wird.

Ergebnis

- 1: Der Operator war erfolgreich.
- -1: Der Operator ist fehlgeschlagen.

Ursache

- "Abgeschlossen", wenn der Operator erfolgreich abgeschlossen wurde.
- Eine Fehlermeldung wird zurückgegeben, wenn der Operator fehlgeschlagen ist.

remoteSSHHost

PORT

LocalPort

SendProtocol

ConnTimeout

CommandData

IsReadData

ReadDataPeriod

ReadDataBytes

ReplyPattern

StartTime

StartDate

Operator-Ports

Erfolg

- Der Operator wurde erfolgreich abgeschlossen.
- Wenn der Operator zum Lesen der Daten aus dem Service konfiguriert wird, ist der Operator erfolgreich wenn:
  - TCP:

    a. An den lokalen Port (wenn angegeben) gebunden ist.

    b. Eine Verbindung zum Remote-Host am angegebenen Remote-Port hergestellt wird.

    c. Daten in den Service (wenn angegeben) geschrieben werden.

    d. Die Antwortdaten aus dem Service gelesen werden.

    e. Wenn sichergestellt wird, dass die Antwortdaten mit dem Muster (wenn angegeben) übereinstimmen.
  - UDP:

    a. An den lokalen Port (wenn angegeben) gebunden ist.

    b. Eine UDP-Meldung zum Remote-Host am angegebenen Remote-Port gesendet wird.

    c. Die Antwort aus dem Service gelesen wird.

    d. Wenn sichergestellt wird, dass die Antwortdaten mit dem Muster (wenn angegeben) übereinstimmen.
- Wenn der Operator so konfiguriert wird, dass keine Daten aus dem Service gelesen werden, ist der Operator erfolgreich wenn:
  - TCP:

    a. An den lokalen Port (wenn angegeben) gebunden ist.

    b. Eine Verbindung zum Remote-Host am angegebenen Remote-Port hergestellt wird.

    c. Daten in den Service (wenn angegeben) geschrieben werden.
  - UDP:

    a. An den lokalen Port (wenn angegeben) gebunden ist.

    b. Eine UDP-Meldung zum Remote-Host am angegebenen Remote-Port gesendet wird.

    Dieser Prozess bedeutet nicht, dass der UDP-Service einsatzbereit ist. Wir empfehlen, den Operator zum Lesen der Antwort aus dem Service zu konfigurieren und mit dem Muster übereinzustimmen. Dann können Sie sicher sein, dass der UDP-Service aktiv ist und ausgeführt wird. Sie können nicht sicher sein, dass ein Service durch UDP ordnungsgemäß überprüft wird, wenn dieser Service keine Daten zurückgibt.

Zeitlimit

Zeitlimit tritt ein, wenn versucht wird, über TCP eine Verbindung zum Service herzustellen. Der Wert für das Zeitlimit wird in Zeitlimit für Verbindung (Sek.) (siehe Seite 365) angegeben.

Fehlgeschlagen

Zu den Gründen gehören:

- Der Operator wird zum Lesen der Daten aus dem Service eingestellt, aber die gelesenen Daten stimmen nicht mit dem Muster überein, die der Anwender angegeben hat.
- Der Anwender gibt ungültige Daten an. Zum Beispiel: negative Remote-Ports bzw. lokale Ports, negatives Zeitlimit für Verbindung (nur für TCP), 0 oder negativer Wert für "Zeit bis zum Lesen von Daten (Sek.)".
- Der Remote-Host ist unbekannt.
- Über TCP kann keine Verbindung zum Remote-Host am Remote-Port hergestellt werden (oder die Verbindung wird abgelehnt). Der Service könnte in diesem Fall deaktiviert sein.
- Es kann nicht zum angegebenen lokalen Port gebunden werden.
- Fehler beim Senden oder Empfangen der Daten in den bzw. aus dem Service.

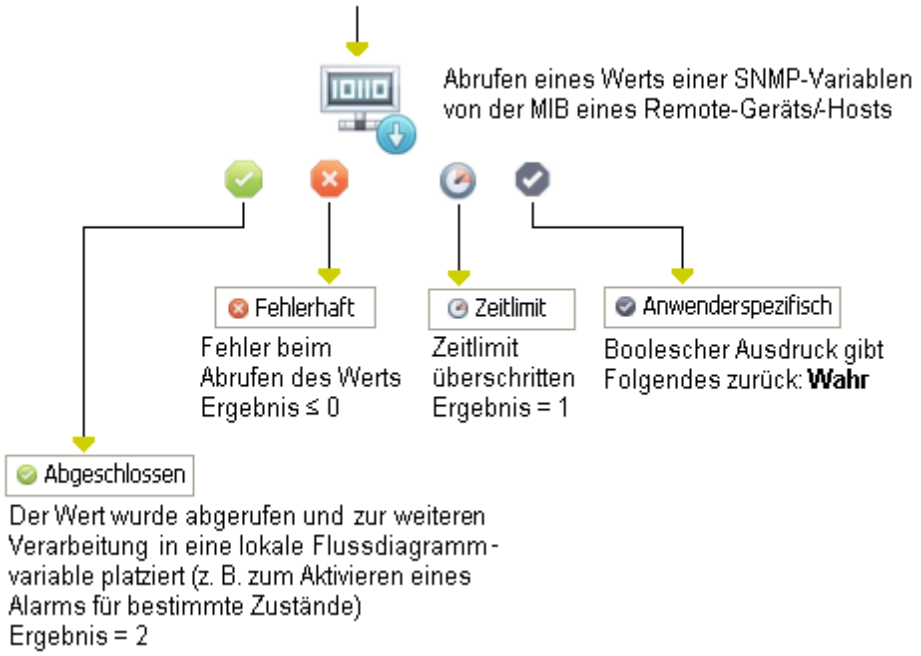
Anwenderspezifische Ports

Wird zurückgegeben, wenn während des Prozessdesigns durch den Anwender festgelegt.

Operator "SNMP-Variable abrufen"

Der Operator "SNMP-Variable abrufen" liest den Wert einer Remote-MIB (Management Information Base)-Variablen. Die Objekt-IDs (OIDs) und Semantik von SNMP-Variablen werden im MIB des Remote-SNMP-Agenten dokumentiert.

SNMP ist ein verbindungsloses, unzuverlässiges Protokoll. Eine Zeitlimit-Option gibt den zulässigen Zeitraum an, innerhalb dessen die Anfrage die Zieladresse erreichen muss. Eine Wiederholungsoption gibt an, wie oft eine Anfrage in Fall von Fehlern gesendet wird. Dieser Operator kann aufgrund verschiedener Fehlerbedingungen fehlschlagen, z. B. wenn die SNMP-Variable nicht gefunden wird, die Leseberechtigung verweigert wird oder ein Gerät nicht verfügbar ist.


Abrufen eines Werts einer SNMP-Variablen von der MIB eines Remote-Geräts/-Hosts
Fehlerhaft
Fehler beim Abrufen des Werts Ergebnis ≤ 0
Zeitlimit
Zeitlimit überschritten Ergebnis = 1
Anwenderspezifisch
Boolescher Ausdruck gibt Folgendes zurück: Wahr
Abgeschlossen
Der Wert wurde abgerufen und zur weiteren Verarbeitung in eine lokale Flussdiagramm-variable platziert (z. B. zum Aktivieren eines Alarms für bestimmte Zustände) Ergebnis = 2

Eingabeparameter

Agentenhost

Definiert die IP-Adresse oder den vollqualifizierten Domänennamen für den Agenten-Host. Zum Beispiel: 192.168.1.254.

Um einen Port mit dem Hostnamen anzugeben, verwenden Sie eines der folgenden Formate:

- Host:Port
- Host/Port

Zum Beispiel: comet.hq.company.com:10162

Community

Definiert die Community, unter der auf die Variable (z. B. "public") zugegriffen werden soll.

ID ldap OID {

Definiert die Objekt-ID (OID) für die Variable. OIDs sind in der Management Information Base (MIB), die einem Remote-Agent zugeordnet ist, dokumentiert. Zum Beispiel: 1.3.6.1.2.1.1.1.0.

Anzahl der Wiederholungen

Gibt an, wie oft das Produkt eine fehlgeschlagene Anfrage wiederholt.

Zeitlimit-Intervall (Sek.)

Definiert das Intervall bis zum Ablaufen des Operators in Sekunden.

SNMP Version (Produktversion)

Gibt die Versionsnummer für den SNMP-Agent an. Wählen Sie "**Version 1**" oder "**Version 2**" aus der Liste aus.

Ausgabeparameter

Object_ID

Retry_Count

Zeitlimit

SNMP_Version

Community

Agent_Host

Port

OIDValue

Agent_Host

Community

RequestId

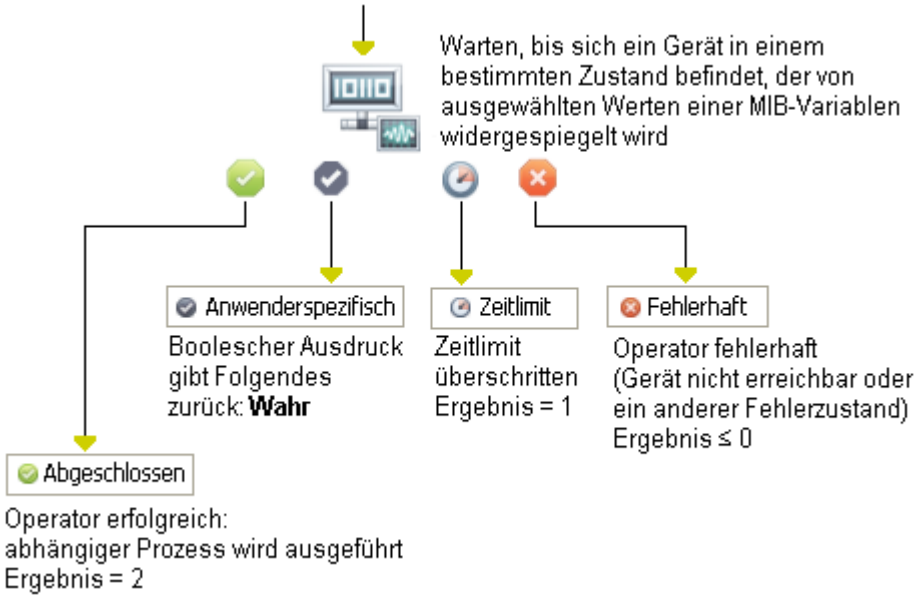
ErrorIndex

ErrorStatus

Operator "SNMP-Variable überwachen"

Der Operator "SNMP-Variable überwachen" wartet, bis eine SNMP-Variable einen Wert hat, der die angegebenen Bedingungen erfüllt. Diesen Bedingungen werden durch ein Muster oder einen Wertebereich definiert. Der Operator kann Teilzeichenfolgen aus einem übereinstimmenden Muster extrahieren.

Dieser Operator wird mit einem iterativen Operator "SNMP-Variable abrufen" implementiert, bis die angegebene Bedingung erfüllt wird.



Eingabeparameter

Agentenhost

Definiert die IP-Adresse oder den vollqualifizierten Domänennamen für den Agenten-Host. Zum Beispiel: 192.#68.1.254

Um einen Port mit dem Hostnamen anzugeben, verwenden Sie eines der folgenden Formate:

- Host:Port
- Host/Port

Zum Beispiel: comet.hq.company.com:10162

Community

Definiert die Community, unter der auf die Variable (z. B. "public") zugegriffen werden soll.

ID ldap OID {

Definiert die Objekt-ID (OID) für die Variable. OIDs sind in der Management Information Base (MIB), die einem Remote-Agent zugeordnet ist, dokumentiert. Zum Beispiel: 1.3.6.1.2.1.1.1.0

Anzahl der Wiederholungen

Gibt an, wie oft das Produkt eine fehlgeschlagene Anfrage wiederholt.

Zeitlimit-Intervall (Sek.)

Definiert das Intervall in Sekunden, die das Produkt auf eine Variable "Get SNMP" wartet, bevor das Zeitlimit des Operators überschritten ist.

Variablentyp

Gibt den Datentyp für die Variable an. Um diesen Parameter zu konfigurieren, wählen Sie einen Wert aus der Drop-down-Liste aus.

Ruhezeit (Sek.)

Definiert das größtmögliche Intervall zwischen zwei Versuchen, den Wert der überwachten Variablen zu überprüfen, in Sekunden.

SNMP Version (Produktversion)

Gibt die Versionsnummer für den SNMP-Agent an. Wählen Sie "**Version 1**" oder "**Version 2**" aus der Liste aus.

Mindestwert

Definiert den niedrigsten erwarteten Zahlenwert.

Höchstwert

Definiert den höchsten erwarteten Zahlenwert.

Maske

Definiert das Muster, nach dem der Operator sucht (weitere Informationen finden Sie im *Handbuch für Inhaltsdesign*).

Trennzeichen (#) im Muster begrenzen Texte, die in Datensatzvariablen des Operators gespeichert werden sollen.

Trennzeichen

Definiert das Zeichen, das die Zone begrenzt, in der die durch Variablennamen angegebenen Variablen gespeichert werden sollen.

Variablennamen

Definiert die Variablennamen, in denen Text gespeichert werden soll, der mit den abgegrenzten Zonen im Muster übereinstimmt. Das Produkt speichert abgegrenzte Zonen sortiert in den aufgelisteten Variablen. Das Produkt verwendet den Schlüsselwortprozess, um über den Prozessdatensatz auf Operatordatensatzvariablen zuzugreifen.

Wenn zum Beispiel die Variablennamen "V1" und "V2" angegeben sind, werden den Operatordatensatzvariablen "Process.Operator_name.V1" und "Process.Operator_name.V2" extrahierte Teilzeichenfolgen zugewiesen. Um die Variablen zum Speichern der übereinstimmenden Zeichenfolgen hinzuzufügen, zu entfernen und zu ordnen, verwenden Sie die Symbolleiste.

Musterübereinstimmung mit berücksichtigter Groß- bzw. Kleinschreibung

Gibt an, ob beim Abgleichen mit dem Muster Groß- und Kleinschreibung berücksichtigt werden soll.

Aktiviert: Groß- bzw. Kleinbuchstaben in einem Muster sollen jeweils nur mit Groß- bzw. Kleinbuchstaben übereingestimmt werden.

Deaktiviert: Groß- und Kleinschreibung soll beim Abgleichen des Musters ignoriert werden.

Ausgabeparameter

Object_OID

Retry_Count

Zeitlimit

Variable_Type

Sleep Time

SNMP_Version

Low_Value

High_Value

Maske

Seperator

Variable_Names

IsCaseSensitivePatternMatching

Watch Expiration

Community

Agent_Host

Port

MatchedEntry

LastReadPos

RequestId

ErrorIndex

ErrorStatsu

Port

OIDValue

Operator "Host pingen"

Mit dem Operator "Host pingen" können Sie den Zugriff auf einen bestimmten Host oder auf eine bestimmte IP-Adresse auswerten. Sie können die Anzahl der Anfragen an den Remote-Host sowie das Zeitlimit und TTL-Werte angeben. Sie können auch die lokale IP-Adresse des Computers angeben, auf dem der Operator ausgeführt wird. In diesem Fall verwendet der Operator die lokale Netzwerkschnittstelle, die der lokalen IP-Adresse zugeordnet ist, um den Ping-Vorgang zu initiieren.

Die Ausgabevariable: "isHostReachable" zeigt an, ob der Host erreichbar ist.

- Wenn *eine* der Ping-Anfragen anzeigt, dass der Host erreichbar ist, wird "isHostReachable" dann auf "Wahr" festgelegt.
- Wenn *alle* Ping-Anfragen anzeigen, dass der Host nicht erreichbar ist, wird "isHostReachable" dann auf "Falsch" festgelegt und der Operator schlägt fehl.

Der Operator "Host pingen" schlägt fehl, wenn ein Fehler auftritt oder wenn *alle* Ping-Anfragen an einen Host fehlschlagen.

Eingabeparameter

Name des Remote-Host

Gibt den Hostnamen oder die zu pingende IP-Adresse an. Verwenden Sie für die IPv6-Adresse entweder das in RFC 2732 definierte Formular oder das wörtliche in RFC 2373 definierte IPv6-Adressformat. Wenn nicht angegeben, wird der Standardwert verwendet.

Standard

Die Loopback-Adresse des Hosts, der dem Kontaktpunkt zugeordnet ist.

Lokale IP-Adresse

Gibt die lokale IP-Adresse des Hosts mit dem Agenten an, der dem Kontaktpunkt zugeordnet ist, dessen Netzwerkschnittstelle den Ping initiiert. Wenn nicht angegeben, verwendet der Operator "Host pingen" den Standardwert.

Standard

Leer - Zeigt eine beliebige Schnittstelle an.

Anzahl der Anfragen

Gibt die Anzahl an, wie oft der Betriebsablauf, der überprüft, ob der Remote-Host erreichbar ist, ausgeführt werden soll. Der Operator "Host pingen" erachtet den Remote-Host als nicht erreichbar, wenn alle diese Anfragen zurückgeben, dass der Host nicht erreichbar ist. Wenn nicht angegeben, verwendet der Operator den Standardwert.

- Wenn *eine* der Ping-Anfragen anzeigt, dass der Host erreichbar ist, wird "sHostReachable" dann auf "Wahr" festgelegt.
- Wenn *alle* Ping-Anfragen anzeigen, dass der Host nicht erreichbar ist, wird "isHostReachasble" dann auf "Falsch" festgelegt und der Operator "Host pingen" schlägt fehl.

Standard

1

Zeit bis Ablauf

Gibt den größtmöglichen Wert für "Zeit bis Ablauf" für jede Anfrage in der angegebenen Anzahl der Anfragen an. Bei Ping (ICMP-Anfragen) steht er für die Anzahl von Hops für die Pakete, bevor die Anfrage abgebrochen und der Remote-Host als nicht erreichbar erachtet wird. Wenn nicht angegeben, verwendet der Operator den Standardwert.

Standard

30

Zeitlimit (Sek.)

Gibt das Zeitlimit in Sekunden an, wobei der Wert für jede Anfrage in der angegebenen Anzahl der Anfragen gilt. Wenn eine Anfrage das Zeitlimit überschreitet, bevor eine Antwort erhalten wird, erachtet diese Anfrage den Remote-Host als nicht erreichbar. Wenn nicht angegeben, verwendet der Operator den Standardwert.

Standard

5

Ausgabeparameter

isHostReachable

"isHostReachable" wird auf einen der folgenden Werte eingestellt:

Wahr

Zeigt an, dass mindestens eine der Ping-Anfragen den Host erreicht hat.

False

Zeigt an, dass keine der Ping-Anfragen den Host erreicht hat. Der Operator schläft fehl.

Eine Anfrage erachtet den Host als nicht erreichbar, wenn folgende Bedingungen vorliegen:

- Der Host ist nicht erreichbar
- TTL läuft ab
- Zeitlimit läuft ab

HostIpAddress

Die IP-Adresse des Remote-Host.

HostCanonicalName

Der kanonische Name des Remote-Host. Der Operator ruft entweder den FQDN oder die IP-Adresse ab, je nach den zugrunde liegenden Systemkonfigurationen.

LocalNetworkInterface

Eine der folgenden Komponenten:

Der Name der lokalen Netzwerkschnittstelle, die verwendet wird, um die Anfragen an den Remote-Host zu senden.

Zeigt den Anwender an, der in der lokalen IP-Adresse angegeben ist.

Leer

Zeigt an, dass keine lokale IP-Adresse in der Operatoreingabe angegeben wurde.

TotalRequests

Die Anzahl der Anfragen, die über den Operator herausgegeben wurde.

Wenn beim Ausführen des Operators kein Fehler auftritt, sollte "TotalRequests" mit dem Wert von "Anzahl der Anfragen" identisch sein.

SuccessfulRequests

Die Anzahl der Anfragen, die den Remote-Host erreicht haben.

FailedRequests

Die Anzahl der Anfragen, die den Remote-Host als nicht erreichbar vorgefunden haben.

FailurePercentage

FailedRequests * 100 / TotalRequests.

Ergebnis

1

Zeigt an, dass der Operator erfolgreich abgeschlossen wurde.

-1

Zeigt an, dass der Operator fehlschlagen ist.

Ursache

Eine der folgenden Komponenten:

Abgeschlossen

Zeigt an, dass der Operator erfolgreich abgeschlossen wurde.

<error message>

Gibt in einer Fehlermeldung an, warum der Operator fehlgeschlagen ist.

remoteSSHHost

localIp

NUMBER_OF_REQUESTS

TTL

Zeitlimit

Operator-Ports

Erfolg

Der Operator wurde erfolgreich abgeschlossen.

Fehlgeschlagen

Der Operator schlägt aus einem der folgenden Gründe fehl:

- "isHostReachable" ist "Falsch".
- Eine Firewall oder ein Netzwerkproblem verhindert, dass CA Process Automation eine Suche startet oder eine Verbindung zum Computer herstellt.
- Unbekannter Name des Remote-Host.
- Ungültige lokale IP-Adresse.
- Die lokale Netzwerkschnittstelle, die der lokalen IP-Adresse zugeordnet ist, kann nicht abgerufen werden.
- Die lokale Netzwerkschnittstelle, die der lokalen IP-Adresse zugeordnet ist, ist nicht aktiviert.
- "Zeitlimit", "Anzahl der Anfragen" oder "Zeit bis Ablauf " ist weniger oder gleich 0 (null).
- Ein anderer Grund, der im Feld "Grund" angegeben ist.

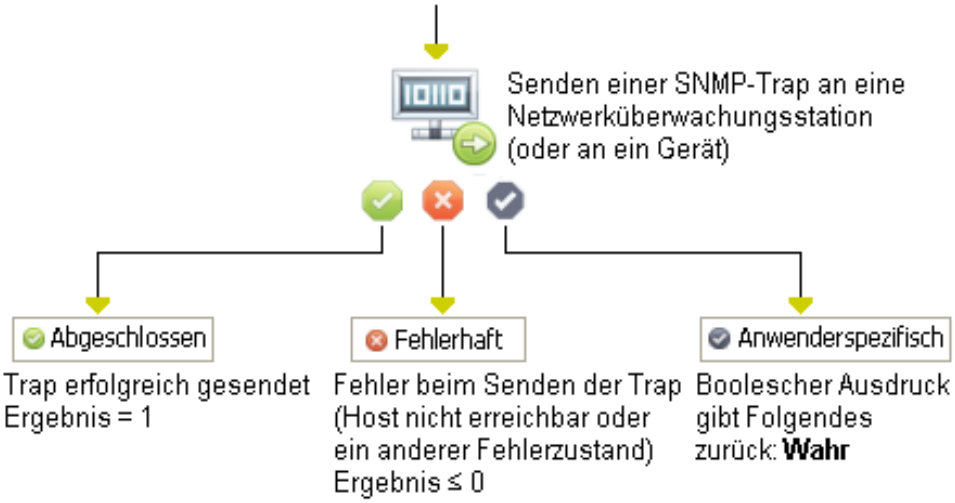
Anwenderspezifische Ports

Wenn während des Prozessdesigns durch den Anwender festgelegt.

Operator "SNMP-Trap senden"

Der Operator "SNMP-Trap senden" generiert SNMP-Traps. Eine *Trap* ist eine nicht angeforderte Meldung, die ein SNMP-Agent an ein SNMP-Managementsystem sendet. Der Agent sendet eine Trap, wenn er entdeckt, dass ein bestimmter Event-Typ lokal auf dem verwalteten Host eingetreten ist. Zum Beispiel kann der Agent eine Trap-Meldung als Reaktion auf ein Systemstart-Event senden. SNMP-Traps werden normalerweise verwendet, um Alarme und Benachrichtigungen auszulösen oder vordefinierte Aktionen auf Remote-Geräten zu bewirken (z. B. Neustart oder Zurücksetzung eines Geräts).

Die genaue Semantik von bestimmter Traps ist in der Dokumentation der MIB (Management Information Base) des SNMP-Agent definiert. Für anwenderspezifische Traps lesen Sie die Dokumentation des Zielagenten.


Senden einer SNMP-Trap an eine Netzwerküberwachungsstation (oder an ein Gerät)
Abgeschlossen
Trap erfolgreich gesendet
Ergebnis = 1
Fehlerhaft
Fehler beim Senden der Trap (Host nicht erreichbar oder ein anderer Fehlerzustand)
Ergebnis ≤ 0
Anwenderspezifisch
Boolescher Ausdruck gibt Folgendes zurück: Wahr

Eingabeparameter

Agentenhost

Definiert die IP-Adresse oder den vollqualifizierten Domänennamen für den Agenten-Host.

Um einen Port mit dem Hostnamen anzugeben, verwenden Sie eines der folgenden Formate:

- Host:Port
- Host/Port

  Zum Beispiel: comet.hq.company.com: #####

Community

Definiert den Namen der SNMP-Trap-Community. Zum Beispiel: public.

SNMP Version (Produktversion)

Gibt die Versionsnummer für den SNMP-Agent an. Wählen Sie "Version 2" aus der Liste aus.

Trap-ID

Gibt eine der standardmäßigen Trap-IDs an:

- Cold Start
- Eine unternehmensspezifische anwenderspezifische Trap-ID
- Egb Neighbor Loss
- Link Down
- Link Up
- Warm Start
- Authentication Failure

Anwenderspezifische Trap-ID

Wenn Sie den Trap-ID-Parameter auf "Anwenderspezifisch" festlegen, definiert dieser Parameter die unternehmensspezifische anwenderspezifische Trap-ID. Weitere Informationen finden Sie in der Dokumentation des Zielagenten.

Nutzdaten-Trap-ID

Definiert die Nutzdaten-Trap-ID.

Nutzdaten-Trap-Typ

Gibt den Datentyp für die Nutzdaten der SNMP-Trap-Meldung an.

Nutzdaten-Trap-Wert

Gibt den Wert für die Nutzdaten der SNMP-Trap an.

Ausgabeparameter

SNMP_Version

Trap_ID

Custom_Trap_ID

Payload_Trap_ID

Payload_Trap_Type

Payload_Trap_Value

Community

Agent_Host

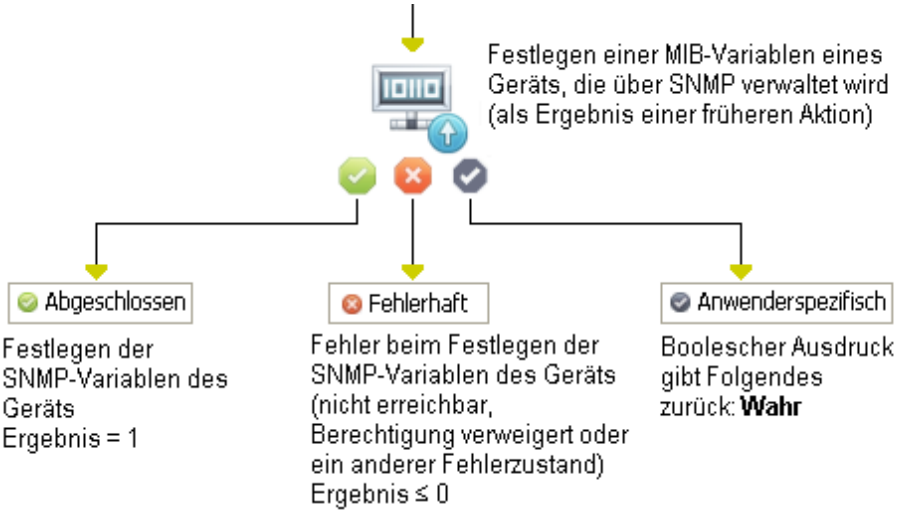
Port

RequestId

Operator "SNMP-Variable aktualisieren"

Der Operator "SNMP-Variable aktualisieren" legt den Wert einer von einem Remote-SNMP-Agenten verwalteten SNMP-Variablen fest. Im Allgemeinen steuern SNMP-Variablen das Verhalten von IP-Geräten. Die genaue Semantik von SNMP-Variablen wird in dem zum Gerät gehörigen MIB angegeben.

Zum Festlegen der Variable muss das Anwenderkonto, das den Operator "SNMP-Variable aktualisieren" ausführt, über Schreibberechtigung auf dem SNMP-Server verfügen, um den Wert des OID ändern zu können.


Festlegen einer MIB-Variablen eines Geräts, die über SNMP verwaltet wird (als Ergebnis einer früheren Aktion)
Abgeschlossen
Festlegen der SNMP-Variablen des Geräts
Ergebnis = 1
Fehlerhaft
Fehler beim Festlegen der SNMP-Variablen des Geräts (nicht erreichbar, Berechtigung verweigert oder ein anderer Fehlerzustand)
Ergebnis ≤ 0
Anwenderspezifisch
Boolescher Ausdruck gibt Folgendes zurück: Wahr

Eingabeparameter

Agentenhost

Gibt die IP-Adresse oder den vollqualifizierten Domänennamen für den Agentenhost an. Zum Beispiel: 192.#68,1.254.

Sie können einen Port zusammen mit dem Hostnamen in einem der folgenden Formate angeben:

- Host:Port
- Host/Port

Community

Gibt die Community an, unter der auf die Variable zugegriffen werden kann (z. B. public).

ID ldap OID {

Gibt die Objekt-ID für die Variable an. Objekt-IDs (OIDs) sind in der MIB (Management Information Base) dokumentiert, die einem Remote-Agenten zugeordnet ist.

Variablentyp

Gibt den Datentyp für die Variable an. Wählen Sie einen der Listenwerte aus, um diesen Operatorparameter zu konfigurieren.

Variablenwert

Gibt den Wert für die Variable an.

SNMP-Version

Gibt den Versionsnummer für den Remote-SNMP-Agenten an. Wählen Sie "Version 1" oder "Version 2" aus der Liste aus.

Ausgabeparameter

Object_ID

Variable_Type

Variable_Value

SNMP_Version

Community

Agent_Host

Port

RequestId

ErrorIndex

ErrorStatus

Kapitel 13: Prozesssteuerung

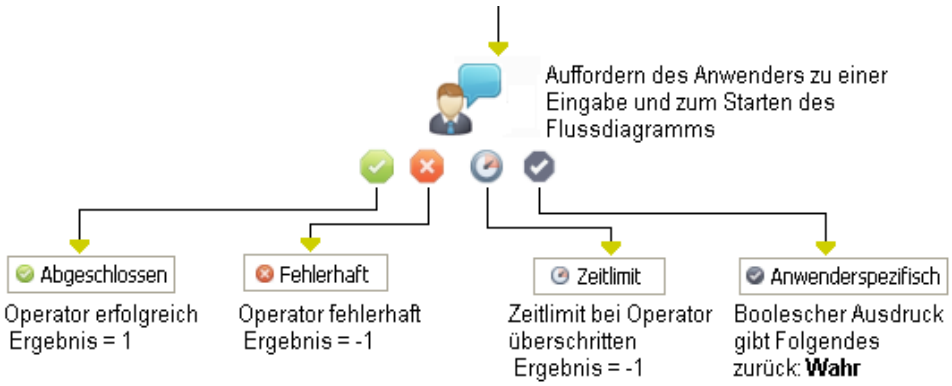
Die Prozesssteuerungs-Operatoren führen CA Process Automation-Prozesse aus und überwachen und steuern sie. Die Prozesssteuerungs-Operatoren starten Systemprozesse aus einem Prozess heraus. Der Operator "Anwenderaufgabe zuweisen" fordert Anwender während der Ausführung eines Prozesses zur Eingabe von Informationen auf.

Alle Prozesssteuerungs-Operatoren können nur auf Koordinationsrechnern ausgeführt werden, nicht auf Agenten.

Operator "Anwenderaufgabe zuweisen"

Der Operator "Anwenderaufgabe zuweisen" zeigt ein CA Process Automation-Interaktionsauftragsformular an, um einen Anwender zu einer Eingabe aufzufordern, und wartet auf die Eingabe. Auf der letzten Seite des Dialogfelds "Interaktionsauftragsformular" wird der Anwender aufgefordert, das Interaktionsauftragsformular zu genehmigen oder abzulehnen. Sie können ein Zeitlimit angeben oder unbeschränkt lange warten. Die Anwendereingabewerte werden in Variablen im Operatordatensatz "Anwenderaufgabe zuweisen" gespeichert.

Sie können den Operator "Anwenderaufgabe zuweisen" auch verwenden, um einem Anwender mitzuteilen, dass auf eine bestimmte Aufgabe über eine eingebettete URL geantwortet werden soll.



Der Operator "Anwenderaufgabe zuweisen" verfügt standardmäßig über vier Ausgangs-Links (nicht anwenderspezifisch):

Abgeschlossen

Wird verarbeitet, wenn ein Anwender folgende Aktionen ausführt:

- Das Interaktionsauftragsformular wird erfolgreich ausgeführt.
- Die Anwendereingabeaufforderung auf der letzten Seite des Interaktionsauftragsformulars wird genehmigt.

Die Parameterwerte werden den Variablen des Operatordatensatzes zugewiesen. Die Variable "Ergebnis" wird auf "1" gesetzt, und die Variable "Grund" wird auf "COMPLETED" gesetzt. Die Aufgabe wird in der Aufgabenliste auf der Registerkarte "Vorgänge" im Zustand "Abgeschlossen" angezeigt.

Abgebrochen

Wird in den folgenden Fällen verwendet:

- Für einen Prozess hat wird der Operator "Anwenderaufgabe zuweisen" ausgeführt, die Aufgabe wird in der Aufgabenliste angezeigt, und die Antwort eines Anwenders ist ausstehend. Bevor jemand antwortet, stößt der Prozess auf einen Operator "Bei Fehlschlag beenden" (normalerweise, da ein anderer Pfad im Prozess noch ausgeführt wird, während "Anwenderaufgabe zuweisen" auf eine Anwenderantwort wartet).
- Der Operator "Anwenderaufgabe zuweisen" wird abgebrochen (manuell oder weil der Prozess geendet hat, während der Operator noch ausgeführt wurde).
- Ein Prozess wurde unterbrochen, und der Operator "Anwenderaufgabe zuweisen" hat die Aufgabe abgebrochen.
- Ein Prozess wurde unterbrochen, und der Operator "Anwenderaufgabe zuweisen" wird zurückgesetzt (manuell oder automatisch).
- Eine Prozessausführung überspringt die Auswahl, die in der Anwendereingabeaufforderung getroffen wurde (im Feld "Formularanordnung").

Die Variable "Ergebnis" wird auf "-1" gesetzt, und die Variable "Grund" wird auf "CANCELED" gesetzt. In allen Fällen wird die Aufgabe in der Aufgabenliste auf der Registerkarte "Vorgänge" im Zustand "Abgebrochen" angezeigt.

Zeitlimit

Wird verarbeitet, wenn der Anwender das Interaktionsauftragsformular innerhalb keines optional angegebenen Zeitlimit-Intervalls fertigstellt. Alle Parameterwerte werden den Variablen des Operatordatensatzes zugewiesen. Die Variable "Ergebnis" wird auf "-1" gesetzt, und die Variable "Grund" wird auf "TIMEOUT" gesetzt. Die Aufgabe wird in der Aufgabenliste auf der Registerkarte "Vorgänge" im Zustand "Zeitlimit" angezeigt.

Abgelehnt

Wird verwendet, wenn der Anwender in der Anwendereingabeaufforderung (im Feld "Formularanordnung") "Ablehnen" auswählt. Die Variable "Ergebnis" wird auf "_" gesetzt, und die Variable "Grund" wird auf "REJECTED" gesetzt. Die Aufgabe wird in der Aufgabenliste auf der Registerkarte "Vorgänge" im Zustand "Abgelehnt" angezeigt.

Eingabeparameter

Der Operator "Anwenderaufgabe zuweisen" enthält folgende Eingabeparameter.

Parameter von "Verantwortliche"

Die Parameter "Verantwortliche" geben autorisierte CA Process Automation-Anwender oder -Gruppen an, um die Anwendereingabeaufforderung zu genehmigen oder abzulehnen. Der Operator "Anwenderaufgabe zuweisen" überprüft die Anmeldeinformationen der Anwender nur dann, wenn ein Anwender oder Gruppe angegeben ist.

Anwender

Gibt die Namen autorisierter CA Process Automation-Anwender an, die die Anwendereingabeaufforderung genehmigen oder ablehnen können. Trennen Sie mehrere Anwender jeweils durch einen Doppelpunkt (:). Zum Beispiel: andre:thomas:markus:uwe.

Um das Dialogfeld "Anwender" zu öffnen, klicken Sie auf . Wählen Sie einzelne Anwender aus der Liste "Verfügbare Anwender" aus, um die Liste "Ausgewählte Anwender" mithilfe der Pfeilschaltflächen zu verschieben und umgekehrt. Sie können auch einen Anwendernamen eingeben, der im Textfeld gesucht werden soll.

Gruppen

Gibt die Namen autorisierter CA Process Automation-Gruppen an, die die Anwendereingabeaufforderung genehmigen oder ablehnen können. Trennen Sie mehrere Gruppen jeweils durch einen Doppelpunkt (:). Zum Beispiel: domainadmin:pamuser:envconfigadmin (oder bei einem Upgrade: domainadmin:pamuser:envconfigadmin).

Um das Dialogfeld "Gruppen" zu öffnen und Gruppen auszuwählen, klicken Sie auf . Wählen Sie einzelne Gruppen aus der Liste "Verfügbare Gruppen" aus, um die Liste "Ausgewählte Gruppen" mithilfe der Pfeilschaltflächen zu verschieben und umgekehrt. Sie können auch einen Gruppennamen eingeben, der im Textfeld gesucht werden soll.

Parameter von Transfer/Delegates

Die Parameter von Übertragen/Delegierte geben einzelne Anwender an, die von CA Process Automation-Anwendern oder Gruppen autorisiert werden, die Anwendereingabeaufforderung zu genehmigen oder abzulehnen. Der Operator "Anwenderaufgabe zuweisen" überprüft die Anmeldeinformationen der Anwender nur dann, wenn ein Anwender oder Gruppe angegeben ist.

Anwender

Gibt die Namen autorisierter CA Process Automation-Anwender an, die die Anwendereingabeaufforderung genehmigen oder ablehnen können. Trennen Sie mehrere Anwender jeweils durch einen Doppelpunkt (:). Zum Beispiel: andre:thomas:markus:uwe.

Um das Dialogfeld "Anwender" zu öffnen, klicken Sie auf . Wählen Sie einzelne Anwender aus der Liste "Verfügbare Anwender" aus, um die Liste "Ausgewählte Anwender" mithilfe der Pfeilschaltflächen zu verschieben und umgekehrt. Sie können auch einen Gruppennamen eingeben, der im Textfeld gesucht werden soll.

Gruppen

Gibt die Namen autorisierter CA Process Automation-Gruppen an, die die Anwendereingabeaufforderung genehmigen oder ablehnen können. Trennen Sie mehrere Gruppen jeweils durch einen Doppelpunkt (:). Zum Beispiel: domainadmin:pamuser:envconfigadmin (oder bei einem Upgrade: domainadmin:pamuser:envconfigadmin).

Um das Dialogfeld "Gruppen" zu öffnen und Gruppen auszuwählen, klicken Sie auf . Wählen Sie einzelne Gruppen aus der Liste "Verfügbare Gruppen" aus, um die Liste "Ausgewählte Gruppen" mithilfe der Pfeilschaltflächen zu verschieben und umgekehrt. Sie können auch einen Gruppennamen eingeben, der im Textfeld gesucht werden soll.

Anwenderaufgabenparameter

Titel

Gibt einen Titel für die Anwenderaufgabe an (optional). Diese Zeichenfolge beschreibt den Titel des dem Anwender zu präsentierenden Formulars.

Beschreibung

Bietet eine optionale Beschreibung für die Anwenderaufgabe.

Interaktionsauftragsformular

Gibt den Pfad in der CA Process Automation-Bibliothek für das Objekt "Interaktionsauftragsformular" an, das den Anwender zur Eingabe auffordert. Das Interaktionsauftragsformular muss sich in derselben Bibliothek befinden wie der Prozess. Zum Beispiel: /Sicherungen/Formulare/Eingabe.

Um ein Interaktionsauftragsformular in der CA Process Automation-Bibliothek zu suchen, klicken Sie auf .

Um das Interaktionsauftragsformular im Formular-Browser anzuzeigen, wenn eins ausgewählt wurde, klicken Sie auf "Öffnen".

Initialisierungscode für Formulardaten

Hiermit können Sie Code hinzufügen, der Formularfelder zur Laufzeit dynamisch initialisiert. Auf diese Weise können Sie Informationen in schreibgeschützten Feldern anzeigen oder Standardwerte für editierbare Felder ändern.

Um CA Process Automation-Ausdrücke hinzuzufügen zum Ändern von Feldwerten, öffnen Sie ein Code-Editor-Fenster.

Verwenden Sie im Editor das Schlüsselwort "Form", um auf Variablen des Operatordatensatzes im folgenden Format zuzugreifen:

Form.*Feldname*

Erläuterung:

Feldname repräsentiert den Namen des Feldes in der Anwenderinteraktionsformular-Objektdefinition. Beispiel:

```
Form.DateTomorrow = System.tomorrow;
```

Für den aktuellen Operator.

Hinweis: Weitere Informationen zum Erstellen von berechneten Ausdrücken in CA Process Automation finden Sie unter "Eingabeparameter für den Operator "JavaScript ausführen" (siehe Seite 438).

Sie können dieses Feld auch verwenden, um Formularfelder zur Laufzeit dynamisch zu initialisieren. Die Initialisierung ist für einfache und komplexe Typen nicht gleich.

- Bei einfachen Typen: Wenn das Interaktionsauftragsformular ein einfaches Element (Textfeld) "Var_0" hat, kann dieses Element durch Folgendes initialisiert werden:

  ```
  Form.Var_0='text'
  ```

- Bei einem ValueMap: Wenn das Interaktionsauftragsformular ein ValueMap hat, das "ID value_map" mit einem Textfeld mit "ID text_field" enthält, muss das "text_field" folgendermaßen initialisiert werden:

  ```
  Form.value_map= newValueMap();
  Form.valuemap.text_field="test";
  ```

- Wenn das Interaktionsauftragsformular eine geschachtelte ValueMap hat, insbesondere wenn eine ValueMap in einem ValueMap mit "ID value_map_nested" vorhanden ist, und wenn eine Variable in der gleichen geschachtelten ValueMap mit "ID text_field_nested" vorhanden ist, dann muss die Initialisierung sein:

  ```
  Form.value_map.value_map_nested= newValueMap();
  Form.value_map.value_map_nested.text_field_nested="test";
  ```

Genehmigungsseite anzeigen

Ist diese Option ausgewählt, können Sie die Aufgabe annehmen oder ablehnen. Ist diese Option ausgewählt, wird das eingeschlossene Formular mit einer Seite zum Genehmigen/Ablehnen zum Beantworten am Ende angezeigt, um über das endgültige Ergebnis des Formulars zu entscheiden.

Ausgabeparameter

Titel

Beschreibung

inputForm

showAcceptanceScreen

Userinfo

- InitialAssignedUsers
- initialAssignedGroups
- assignedUsersFilter
- assignedGroupsFilter

Genehmigen

RepliedBy

Task ID

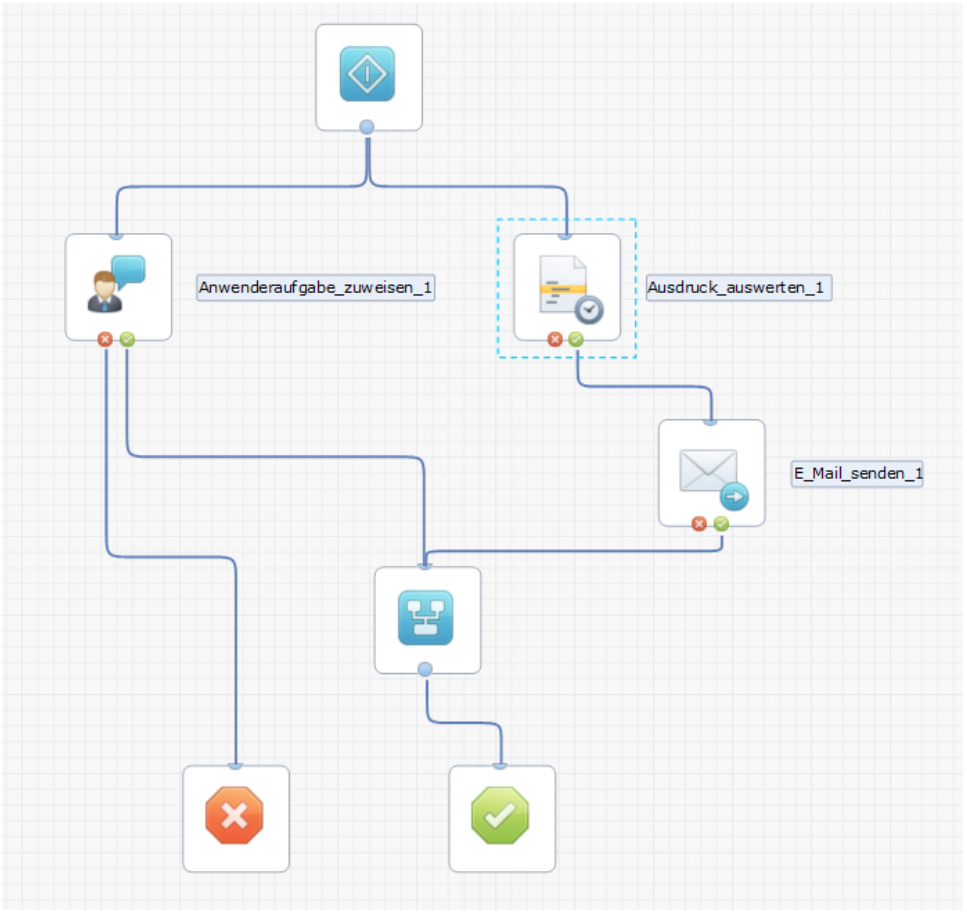
initCode

dueDateTime

Beispiel

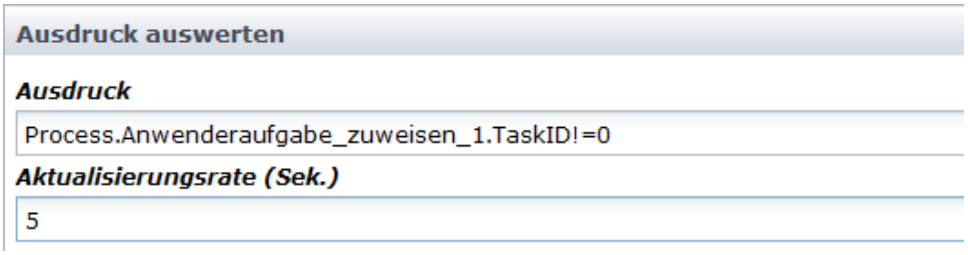
Dieses Beispiel erklärt, wie Sie eine Benachrichtigung an einen Anwender senden können, um mithilfe einer eingebetteten URL in einer E-Mail auf eine Aufgabe zu antworten. Sie können den Ausgabeparameter "Aufgaben-ID" des Operators "Anwenderaufgabe zuweisen" in der eingebetteten URL einschließen, um auf eine direkte Aufgabe zuzugreifen. Sie können die eingebettete URL in den Operator "E-Mail senden" einschließen, um dem Anwender über E-Mail mitzuteilen, dass auf eine Aufgabe geantwortet werden soll.

"Anwenderaufgabe zuweisen" bleibt im Wartzustand, bis der Anwender antwortet. Sie können parallel einen Benachrichtigungsprozess ausführen, um den Anwender mit der direkten URL zu benachrichtigen, um auf eine Aufgabe zu antworten, die, wie hier angezeigt, vom Operator "Anwenderaufgabe zuweisen" ausgelöst wurde:

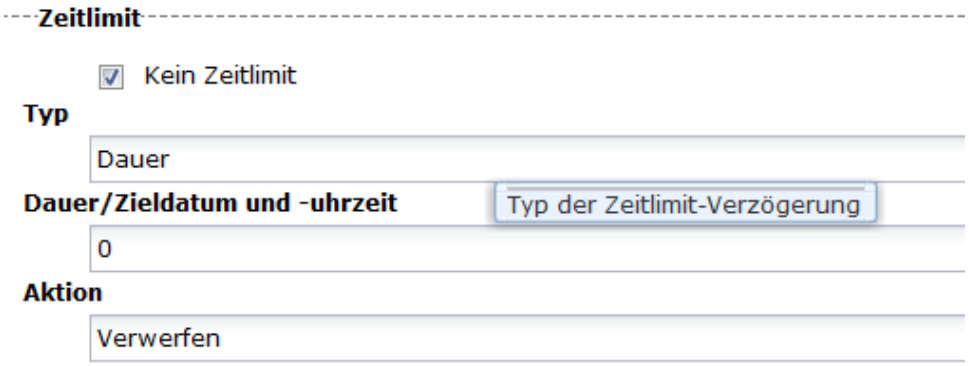


Sie können den Operator "Ausdruck auswerten" verwenden, um zu warten, bis der Ausdruck "Anwenderaufgabe zuweisen" als wahr ausgewertet wird, während der Ausdruck alle fünf Sekunden aktualisiert und auswertet wird.

Die Parameter für "Ausdruck auswerten" können Folgende sein:



Stellen Sie sicher, dass Sie in den Zeitlimit-Parametern von "Anwenderaufgabe zuweisen" das Kontrollkästchen "Kein Zeitlimit" aktivieren, um eine Zeitüberschreitung für diesen Operator zu vermeiden, bis der Ausdruck als wahr ausgewertet wird.



Sobald die Aufgaben-ID dann berechnet und der Wert nicht mehr null ist, wird der Ausdruck als wahr ausgewertet, und der Operator "E-Mail senden" sendet eine E-Mail mit der folgenden eingebetteten URL als Teil der Meldung:

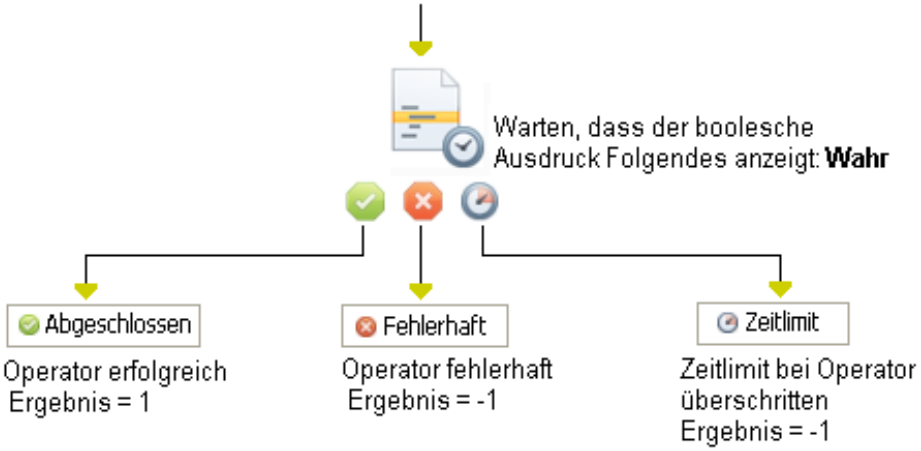
```
getOrchestratorURL() + "itpam" + "?ROID=" + Process.Assign_User_Task_1.TaskID +
"&page=replytask"
```

Der Operator "Und" synchronisiert zwei Verzweigungen des Prozesses in eine einzelne Verzweigung.

Wenn der Anwender die E-Mail erhält, wird die Aufgabe in einer URL dargestellt, auf die der Anwender klickt, um auf die Anmeldeseite von CA Process Automation zu gelangen. Nach der Authentifizierung wird der Anwender direkt zum Formular, das der Aufgabe angehängt ist, weitergeleitet, und der Anwender kann dann direkt auf die bestimmte Aufgabe antworten.

Operator "Ausdruck auswerten"

Der Operator "Ausdruck auswerten" verzögert die Verarbeitung auf der Verzweigung eines Prozesses, bis eine durch einen booleschen Ausdruck repräsentierte Bedingung mit "Wahr" bewertet wird. Dieser Operator bietet einen Mechanismus, um einen Prozess anzuhalten, bis eine Bedingung sich ändert. Der Operator wird oftmals dazu verwendet, um voneinander abhängige Prozesse zu synchronisieren oder die Verwendung von gemeinsam genutzten Ressourcen, die durch Variablen dargestellt werden, zu steuern.



Die Bedingung wird periodisch entsprechend einer angegebenen Rate ausgewertet. Die Rate muss lang genug sein, um die CPU-Auslastung innerhalb annehmbarer Limits zu vergrößern. Wenn eine Bedingung für einen minimalen bekannten Zeitraum vorliegt, kann die Auslastung weiter reduziert werden, indem vor den Operator "Ausdruck auswerten" ein Operator "Verzögerung" (siehe Seite 416) gesetzt wird. Ein Beispiel für eine solche Situation ist, wenn ein anderer Prozess eine Ressource verwendet und sie nicht vor einer bestimmten Tageszeit freigibt.

Eingabeparameter

Ausdruck

Ein boolescher Ausdruck, der eine "Wahr"-Bedingung angibt, wenn einige Bedingungen erfüllt werden. Hier sind zwei Beispiele:

```
(Datasets["/exploit/variables/set_1"].var1 == 1)
```

```
System.Time >= Process.FinishTime
```

Hinweis: Dieses Feld enthält Datensatzunterstützung, wenn ein beliebiges Schlüsselwort verwendet wird.

Aktualisierungsrate (Sek.)

Das Intervall in Sekunden, in dem Bedingung für eine "Wahr"-Bedingung evaluiert wird.

Ausgabeparameter

Ausdruck

RefreshRate

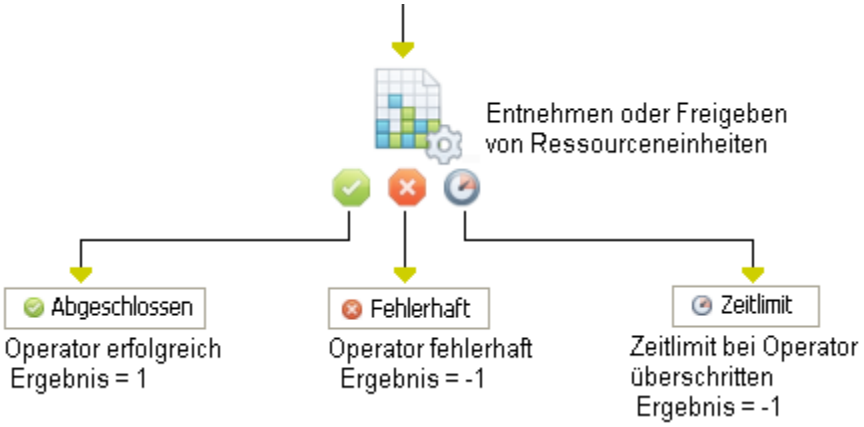
TimeOutSec

TimeOut

Operator "Ressourcen verwalten"

Der Operator "Ressourcen verwalten" führt Aktionen mit CA Process Automation-Ressourcen aus. Zu diesen Aktionen gehören das Übernehmen von Ressourceneinheiten, das Freigeben von Ressourceneinheiten sowie das Sperren und Entsperren von Ressourcen.

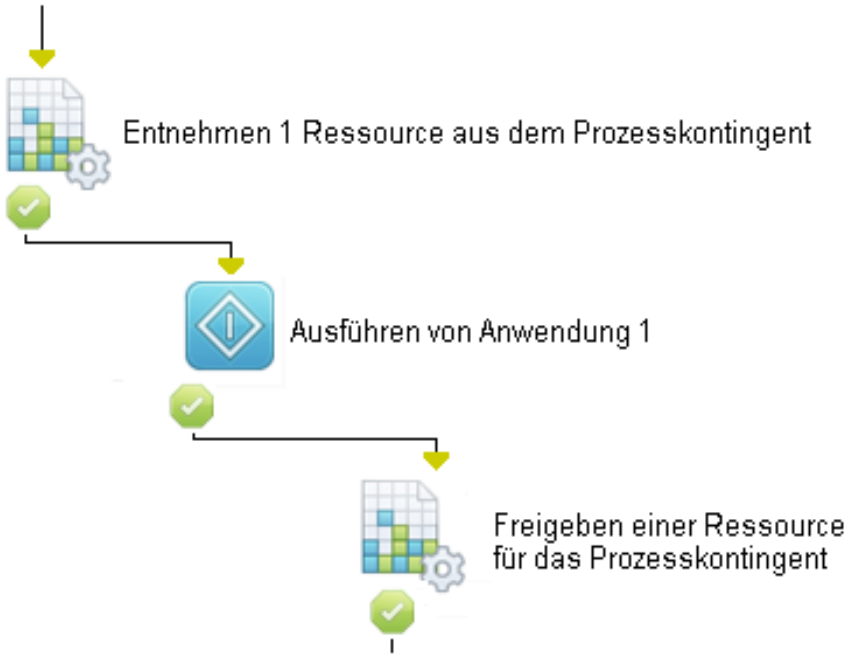
Der Operator "Ressourcen verwalten" bietet eine Möglichkeit, zu überprüfen, ob bestimmte Ressourcen vorhanden sind und darauf zu warten und den Zustand dieser Ressourcen zu beeinflussen. Der Operator kann die Verarbeitung einer Prozessverzweigung von der Ressourcenverfügbarkeit abhängig machen. Innerhalb einer Umgebung können Ressourcenoperatoren verwendet werden, um die Verarbeitung von mehreren Prozessen zu regulieren und zu koordinieren. Die Operatoren stellen sicher, dass individuelle Prozesse ausschließlich Zugriff auf externe Ressourcen haben.



Der Operator "Ressourcen verwalten" verfügt über drei mögliche Ausgangsverknüpfungen:

- "Erfolgreich", wenn Aktionen erfolgreich innerhalb eines angegebenen Zeitlimits ausgeführt werden.
- "Fehlgeschlagen", wenn die Ressourcen nicht vorhanden sind oder eine andere Fehlerbedingung auftritt.
- "Zeitlimit, wenn das angegebene Zeitlimit überschritten wird, bevor die erforderlichen Ressourcen verfügbar werden.

Ressourcen werden normalerweise einem Ressourcenkontingent entnommen, bevor andere Operatoren verarbeitet werden, und wenn die Operatoren abgeschlossen wurden, wieder in das Kontingent eingefügt, wie unten gezeigt:


Entnehmen 1 Ressource aus dem Prozesskontingent
Ausführen von Anwendung 1
Freigeben einer Ressource
für das Prozesskontingent

Bedingungen auf einer Ressource müssen regelmäßig für mögliche Änderungen ausgewertet werden. Achten Sie bei der Angabe des Aktualisierungsintervalls sorgfältig darauf, dass dieses nicht zu kurz ist. Verwenden Sie einen Operator "Datum/Uhrzeit überprüfen" (siehe Seite 197), um die Ausführung des Operators "Ressourcen verwalten" zu verzögern, wenn die benötigten Ressourcen erst verfügbar sind, nachdem:

- Ein angegebener Zeitraum wurde erreicht.

  ODER

- Vor einem bestimmten Datum oder einer bestimmten Uhrzeit.

Eingabeparameter

Aktion

Listet die auszuführenden Aktionen auf. Mit den Schaltflächen "Hinzufügen", "Löschen" und "Bearbeiten" werden Aktionen dieser Liste hinzugefügt, daraus entfernt oder bearbeitet. Jede Aktion gibt Folgendes an:

- Ein Ressourcenobjekt
- Der Name der Ressource im Objekt, der verwendet werden soll

- Die Aktion, die auf der Ressource ausgeführt werden soll (Ressourceneinheiten übernehmen, Einheiten freigeben, eine Ressource sperren oder freigeben)
- Wieviele Einheiten der Ressource übernommen oder freigegeben werden sollen

Aktionseigenschaften

Dieses Dialogfeld definiert eine Aktion, die für eine Ressource ausgeführt werden soll. Klicken Sie neben dem Listenfeld "Aktion" entweder auf die Schaltfläche "Hinzufügen" oder "Bearbeiten". Dieses Dialogfeld öffnet die Ressourceneigenschaften für einen Ressourcenoperator.

ResourcePath

Gibt ein Ressourcenobjekt an. Geben Sie den vollständigen Pfad der Ressource in die CA Process Automation-Bibliothek ein, oder klicken Sie auf , um das Objekt zu finden. Doppelte Anführungszeichen müssen eine Literalzeichenfolge einschließen. Sie können eine Variable "Datensatz" oder einen Ausdruck verwenden, um das Objekt "Ressourcen" anzugeben. Um das Objekt im Ressourcen-Editor zu öffnen, klicken Sie auf die Schaltfläche "Öffnen".

ResourceName

Gibt die Ressource im Ressourcenobjekt an, auf dem die Aktion ausgeführt werden soll. Geben Sie den Namen genau so ein, wie er im Ressourcenobjekt definiert wurde. Doppelte Anführungszeichen müssen eine Literalzeichenfolge einschließen. Sie können eine Datensatzvariable oder einen Ausdruck verwenden, um die Ressource anzugeben.

Aktion

Wählen Sie die für die Ressource auszuführende Aktion aus:

TakeUnits

Übernimmt die Anzahl der im Feld "Menge" festgelegten Ressourceneinheiten.

FreeUnits

Gibt die Anzahl der im Feld "Menge" festgelegten Ressourceneinheiten zurück.

LockResource

Sperrt die Ressource, sodass andere Ressourcen-Operatoren keine Ressourceneinheiten übernehmen und die Ressource nicht sperren können. Diese Aktion erfasst alle nicht verwendeten Ressourceneinheiten für eine Ressource. Aktionen können weiterhin Ressourceneinheiten freigeben, die vor der Sperre einer Ressource übernommen wurden. Allerdings sind die freigegebenen Einheiten nur verfügbar, wenn die Ressource entsperrt ist.

UnlockResource

Entsperrt eine gesperrte Ressource.

Menge

Bei den Aktionen "Einheiten übernehmen" oder "Einheiten freigeben" gibt dieser Wert die Anzahl von zu erfassenden oder freizugebenden Ressourceneinheiten an. "Menge" wird für die beiden anderen Optionen deaktiviert, z. B. "Ressource entsperren" und "Ressource sperren".

Alle Ressourcen müssen verfügbar sein

Wenn diese Option ausgewählt ist, müssen alle Ressourcen, die zur Ausführung der unter "Aktionen" aufgelisteten Aktionen erforderlich sind, innerhalb der Einschränkungen verfügbar sein, die durch die Option "Zeitlimit" festgelegt ist. Der Operator ist nur erfolgreich, wenn alle Ressourcen innerhalb der von der Einstellung "Zeitlimit" festgelegten Zeitlimit-Einschränkungen verfügbar werden.

Wenn diese Option deaktiviert ist, wird der Operator erfolgreich abgeschlossen, wenn die Ressourcen, die für mindestens eine der unter "Aktionen" aufgelisteten Aktionen erforderlich sind, innerhalb der Einschränkungen verfügbar werden, die durch die Option "Zeitlimit" festgelegt sind. Wenn Ressourcen für eine der aufgelisteten Aktionen verfügbar sind, bearbeitet das Prozesssteuerungsmodul die Ausgangsverknüpfung "Erfolgreich" für diesen Operator.

Aktionen ausführen

Bestimmt, ob die unter "Aktionen" aufgelisteten Aktionen ausgeführt werden. Um nur zu überprüfen, ob Ressourcen ohne Ausführen von Aktionen verfügbar sind, deaktivieren Sie dieses Kontrollkästchen. Der Operator führt dann die die Ausgangsverknüpfung "Erfolgreich" aus. Diese Verknüpfung wird *nur* dann ausgeführt, wenn Ressourcen innerhalb der Einschränkungen, die durch die Einstellungen "Zeitlimit" und "Alle Ressourcen müssen verfügbar sein" festgelegt werden, verfügbar sind, ohne eine Aktion auszuführen.

Diese Einstellung kann mit einer Ressource verwendet werden, die so festgelegt wurde, dass ein gesamter Prozesssatz aktiviert oder deaktiviert wird. Diese Prozesse stellen sicher, dass keine Sperre auf der Ressource vorhanden ist, bevor ihre Aufgaben gestartet werden. Diese Überprüfung der Sperre wird durchgeführt, indem versucht wird, eine einzelne Ressourceneinheit aus der Ressource zu erfassen. Je nach Ergebnis des Tests können einige andere Mechanismen die Ressource sperren oder entsperren. Beispiel:

- Ablaufplanaufgaben (bei denen das Aktivieren oder Deaktivieren von Prozessen auf Zeiteinschränkungen basiert)
- Manuell gestartete Aufgaben (mittels eines Startauftragsformulars)
- Ein Prozess, der durch eine externe Überwachungsanwendung gestartet wird (mittels des CA Process Automation-Webservices-Daemons)
- Ein Prozess, der eine interne oder externe Bedingung in einer Schleife überwacht.

Ausgabeparameter

ActionProperties

ResourcePath

ResourceName

Aktion

Menge

Alle Ressourcen müssen verfügbar sein

Aktionen ausführen

TimeOut

Event-Operatoren

CA Process Automation stellt durch zwei Operatoren Event-Verwaltungsfunktionen bereit:

- Event überwachen
- Event senden

Andere Prozesse können Events senden. Zudem können von CA Process Automation bereitgestellte Webservices auch Events senden.

Hinweis: Diese beiden Operatoren werden nur auf Koordinationsrechnern nicht auf Agenten ausgeführt.

Operator "Event überwachen"

Der Operator "Event überwachen" wird in einem Prozess dazu verwendet, auf einen bestimmten Event zu warten, bevor der Ausführungspfad fortgesetzt wird. Zum Beispiel kann ein Prozess auf ein Event warten, das signalisiert, dass ein Ticket genehmigt wurde, statt periodisch das Ticket abzufragen und den Genehmigungsstatus zu überprüfen.

Der Operator "Event überwachen" nutzt die verfügbaren/übereinstimmenden Events als Standardverhalten.

Hinweis: "Event überwachen" kann nicht geplant werden (das heißt, er Operator kann nicht in Ablaufplänen verwendet werden). Allerdings kann ein Anwender einen Prozess mit "Event überwachen" entwerfen und dann den Prozess im Designer für Ablaufpläne planen.

Eingabeparameter

Event-Name

Gibt den Namen des Events an. Dieser Ausdruck wird mit dem Namen des Events abgeglichen. Der Name kann ein regulärer Ausdruck, eine Teilübereinstimmung auf Basis der Anwenderwahl oder beides sein.

Event-Typ

Gibt den Typ des Events an (optional). Dieser Ausdruck wird mit dem Typ des Event abgeglichen. Der Typ kann ein regulärer Ausdruck, eine Teilübereinstimmung auf Basis der Anwenderwahl oder beides sein.

Event-Quelle

Gibt die Quelle des Events an. Dieser Ausdruck wird mit der Quelle des Event abgeglichen. Die Quelle kann ein regulärer Ausdruck, eine Teilübereinstimmung auf Basis der Anwenderwahl oder beides sein.

Event-Ziel

Gibt den Namen des Event-Ziels an (optional). Dieser Ausdruck wird mit dem Ziel des Events abgeglichen. Das Ziel kann ein regulärer Ausdruck, eine Teilübereinstimmung auf Basis der Anwenderwahl oder beides sein.

Ausdruck

Gibt einen CA Process Automation booleschen Ausdruck für zusätzliche Event-Parameter (optional) an. Dieser Ausdruck wird mit dem Feld "Event-Parameter" des Events abgeglichen. Auf Event-Parameter kann über ein "payload"-Schlüsselwort zugegriffen werden (zum Beispiel payload.ticketId=="1443132").

Hinweis: Dieses Feld enthält Datensatzunterstützung, wenn ein beliebiges Schlüsselwort verwendet wird.

Alle übereinstimmenden Events abrufen

Ist diese Option aktiviert, empfängt der Operator "Event überwachen" nicht nur das erste übereinstimmende Event, sondern alle übereinstimmenden Events. Sobald diese Events zugestellt werden, werden sie nicht erneut gesendet. Jedes Event, das an Sie gesendet wurde und mit "An einzelnen Abonnenten senden" gekennzeichnet ist, wird ungültig gemacht und an keine weitere Person gesendet.

Musterübereinstimmung aktivieren

Aktiviert Musterabgleiche mit den entsprechenden Event-Attributen wie "Name", "Typ", "Quelle" und "Ziel".

Teilweise Übereinstimmung erlauben

Lässt eine teilweise Übereinstimmung mit den entsprechenden Event-Attributen wie "Name", "Typ", "Quelle" und "Ziel" zu.

Ausgabeparameter

eventId

eventName

eventType

eventSource

eventDestination

toSingleSubsriber

payload

creationTime

expirationTime

Anwender

Operator "Event senden"

Der Operator "Event senden" wird verwendet, um einen Event beim CA Process Automation-Koordinationsrechner zu veröffentlichen. Events werden vom Event Manager empfangen, der auf dem CA Process Automation-Koordinationsrechner (der alle Abonnenten enthält) ausgeführt wird. Das Event wird auf interessierte Abonnenten hin überprüft, indem die Event-Parameter abgeglichen werden. Alle Abonnenten, die auf diesen Typ von Event warten, werden dann benachrichtigt. Dadurch wird der Operator "Event überwachen" abgeschlossen, und der Prozess wird dem Ausführungspfad entsprechend fortgesetzt.

"Event senden" kann nicht geplant werden (das heißt, der Operator kann nicht in Ablaufplänen verwendet werden). Allerdings kann ein Anwender einen Prozess mit "Event senden" entwerfen und dann den Prozess im Designer für Ablaufpläne planen.

Hinweis: Das gleiche Event wird einem Operator in der gleichen Prozessinstanz niemals zweimal geliefert.

Eingabeparameter

Event-Name

Gibt den Namen des Events an (obligatorisch).

Event-Typ

Gibt den Typ des Events an (optional).

Event-Quelle

Gibt die Event-Quelle an (optional).

Event-Ziel

Gibt das Event-Ziel an (optional).

Abonnenten des Event gleichen einen regulären Ausdruck mit diesen Feldern ab, um zu entscheiden, ob sie an diesem Event interessiert sind.

An einzelnen Abonnenten senden

Ist diese Option auf "wahr" festgelegt, werden die Events nur an genau einen wartenden Prozess gesendet. Das Event wird vom ersten Event-Handler "verbraucht", der momentan Events "verbraucht".

Event-Parameter

Gibt zusätzliche Event-Parameter an, bei denen es sich um einen CA Process Automation-Datentyp handeln kann (optional).

Der "Ausdruck"-Parameter im Operator "Event überwachen" wird gegen "Event-Parameter" ausgewertet. Auf diese Parameter kann über ein payload-Schlüsselwort zugegriffen werden (zum Beispiel payload.ticketId=="1443132").

Läuft ab nach (Sek.)

Gibt an, wie viele Sekunden ein Event für die Abstimmung mit Teilnehmern über die Registerkarte "Designer" benötigen kann.

Ausgabeparameter

expirationDuration

eventId

eventName

eventType

eventSource

eventDestination

toSingleSubsriber

payload

creationTime

expirationTime

Anwender

Verwendungsmuster für Events

Die folgenden zwei Verwendungsmuster sind verfügbar für Events:

Warteschlangenmuster

Jedes Event wird an einen einzigen Verbraucher geliefert. Sie müssen das Event auf der Senderseite dementsprechend markieren (An einzelnen Abonnenten senden). Events dieses Typs werden gelöscht, sobald sie zugestellt wurden oder ablaufen.

Hinweis: Dieses Muster wirkt sich auf Auslöser aus. Weitere Informationen finden Sie im *Handbuch für Inhaltsadministratoren*.

Benachrichtigungsmechanismus

Das Event soll einer beliebigen Anzahl von interessierten Parteien einen Zustand signalisieren. Zum Beispiel zeigt eine Benachrichtigung an, dass sich etwas geändert hat, ein System heruntergefahren wird usw. Ein solches Event wird einmal an alle Abonnenten gesendet, bis das Event-Zeitlimit überschritten wird.

Operator "Prozess starten"

Verwenden Sie den Operator "Prozess starten", um einen Prozess aus einem Prozess heraus zu starten. Der Operator "Prozess starten" erstellt eine neue Instanz eines Prozesses auf einem Kontaktpunkt und stellt eine Startanfrage in die Warteschlange für die entsprechende Engine. Sie können auf den Datensatz für den untergeordneten Prozess über den Operatornamen im Prozessdatensatz für den übergeordneten Prozess verweisen. Verwenden Sie die folgende Syntax:

`Process.`*`Operatorname`*`.`*`Feldname`*

Operatorname steht für den Namen des Operators "Prozess starten" im übergeordneten Prozess.

Feldname ist die Datensatzvariable, auf die Sie im untergeordneten Prozess zugreifen möchten.

Eingabeparameter

Prozessname

Gibt den Pfad für den Prozess in der CA Process Automation-Bibliothek an. Der Prozess muss in der CA Process Automation-Bibliothek des Kontaktpunkts enthalten sein, auf dem der Operator so konfiguriert ist, dass er ausgeführt wird.

Zum Beispiel: "/Doc/NT_Charts/Alert"

Um einen Prozess aus der Bibliothek auszuwählen, klicken Sie auf .

Öffnen

Öffnet den Prozess, der in "Prozessname" zum Bearbeiten angegeben ist. Diese Schaltfläche ist nur verfügbar, nachdem Sie den Pfad zu einem Prozess in das angrenzende Feld eingegeben haben.

Prozessdatensatz-Initialisierungscode

Gibt Anweisungen an, die Datensatzvariablen in dem Prozess initialisieren, der gestartet wird. Beispiel:

- Process.WorkDir = "C:\temp";
- Process.User = Caller.User;
- Process.DatabaseServer = Caller.DatabaseServer;

In diesem Feld verweist das Schlüsselwort *Process* auf den Datensatz in der neuen Instanz des Prozesses, der durch "Prozessname" angegeben ist. Das Schlüsselwort "Caller" verweist auf den Datensatz des Prozesses, der den Operator "Prozess starten" enthält.

Das Schlüsselwort *Process* oder *Caller* ist obligatorisch für ein Verweisen auf oder Erstellen von Variablen im über- oder untergeordneten Prozessdatensatz. Ohne eines dieser Schlüsselwörter geht das Datensatzinitialisierungsskript immer so vor, dass es eine Berechnungsvariable erstellt oder versucht, auf eine solche Variable zu verweisen.

Modus

Wählen Sie eine der folgenden Optionen aus:

Angehängt

Führt den untergeordneten Prozess als einen eigenständigen Prozess aus.

Der Operator "Prozess starten" wird nicht abgeschlossen, bis die Verarbeitung der neuen Instanz beendet wurde. Der Prozess, der den Operator ausführt, ist der übergeordnete Prozess.

Abgetrennt

Führt einen Prozess im abgetrennten Modus aus.

Eine Instanz eines Prozesses, der im abgetrennten Modus gestartet wurde, hat keine übergeordnete Beziehung zu dem Prozess, der ihn gestartet hat, und ist in jeder Aufruffolge, die von diesem Prozess ausgeht, der Stammprozess.

Inline

Führt einen untergeordneten Prozess als einen direkten Teil des übergeordneten Prozesses aus (das heißt, der untergeordnete Prozess wird in den übergeordneten Prozess eingeblendet).

Prozessspurwechsel-Handler vom übergeordneten Prozess erben

Wenn diese Option aktiviert ist, erbt der untergeordnete Prozess den Prozessspurwechsel-Handler vom übergeordneten Prozess (sofern er nicht schon im untergeordneten Prozess definiert ist).

Startdatum

Gibt das Datum an, an dem eine abgetrennte Instanz des Prozesses gestartet werden soll. Der Standardwert ist das Datum, an dem der Operator ausgeführt wird (System.Date). Diese Option ist nur verfügbar, wenn als Prozessmodus die Option "Abgetrennt" ausgewählt ist.

Startzeit

Gibt die Uhrzeit an, zu der eine abgetrennte Instanz des Prozesses gestartet werden soll. Der Standardwert ist die Uhrzeit, zu der der Operator ausgeführt wird (System.Time). Diese Option ist nur verfügbar, wenn als Prozessmodus die Option "Abgetrennt" ausgewählt ist. In Kombination mit der Option "Startdatum" kann mit "Startzeit" ein Prozess so geplant werden, dass er einen anderen Prozess ausführt.

Ausgabeparameter

32WorkflowName

Local (Prozessdatensatz-Initialisierungscode)

processMode (Angehängt, Abgetrennt oder Inline)

inheritLaneChangeFromParentProcess

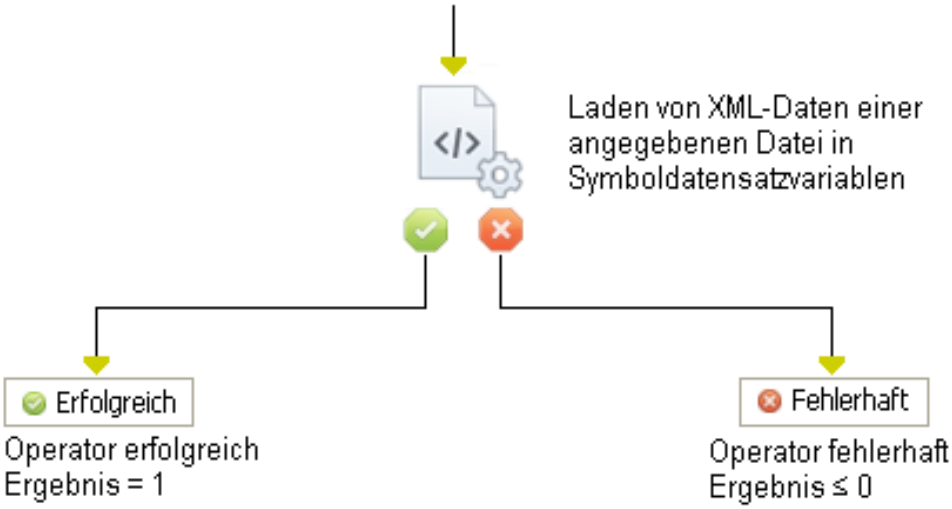
Kapitel 14: Hilfsprogramme

Die Hilfsprogramme-Operatoren können für Hilfsprogrammzwecke in Prozessen verwendet werden.

Operator "XPath anwenden"

Der Operator "XPath anwenden" analysiert Daten und ruft sie aus einem XML-Dokument ab. Dieser Operator unterstützt Folgendes:

- Analysiert ein XML-Dokument und ruft angegebene Daten aus dem Dokument ab.
- Speichert die Ergebnisse in CA Process Automation-Datensätzen, auf die nachfolgende Operatoren in einem Prozess zugreifen.



Eingabeparameter

Eingabequelle

Wählen Sie die Quelle für die SOAP-Dienst-Eingabeanforderung aus: Ausdruck oder Eingabedateiname.

Ausdruck

Geben Sie den Ausdruck an, um XML-Inhalte zu laden. Zum Beispiel:

```
Process.xmlContent
```

oder

```
Datasets["xmlData"].xmlContent
```

XML-Eingabedatei

Gibt das XML-Dokument, aus dem Daten extrahiert werden sollen. Geben Sie einen Ausdruck der XML-Datei für ein gültiges XML-Dokument ein, der den Pfad zurückgibt.

Strip Namespace in XML Structure

CA Process Automation bietet eine Option, um XML-Namespaces aus einer Antwort zu entfernen, sodass ein Anwender einfachere XPath-Ausdrücke angeben kann, um nach dem Wert eines bestimmten Elements zu suchen. Diese Option ist in allen SOAP-Operatoren verfügbar.

Die folgenden JavaScript-Funktionen werden bereitgestellt:

- applyXPath(xmldata,xpath_query,namespaceAware)
- applyXPathToUrl(urls,xpath_query,namespaceAware)

Hinweis: Der Standardwert von "namespaceAware" lautet "wahr". Der Wert von "namespaceAware" ist "falsch", wenn Sie "Namespace" in XML-Struktur entfernen möchten (anderenfalls lautet er "wahr").

```
Process.x="<getMatchingEventsResponse
xmlns='http://www.ca.com/itpam'><events> <event
><eventName>test</eventName></event></events></getMatchingEventsResponse>";
Process.s=applyXPath(Process.x,"//eventName",true);
Process.aa1=applyXPathToUrl("file:C:/test.xml","//message",true);
Process.aa2=applyXPathToUrl("file:C:/test.xml","//message",false);
Process.s1=applyXPath(Process.x,"//eventName",false);
Process.s2=applyXPath(Process.x,"//eventName");
```

Zusätzliche extrahierte Daten

Gibt XPath-Ausdrücke, mit denen Daten aus dem XML-Dokument extrahiert werden. Geben Sie für jeden hier angegebenen Ausdruck eine Datensatzvariable, in der die extrahierten Daten gespeichert werden sollen, und einen Datentyp an.

Verwenden Sie die Schaltflächen "Hinzufügen", "Bearbeiten" und "Löschen", um die Ausdrücke im Listenfeld zu verwalten. Mit den Schaltflächen "Hinzufügen" und "Bearbeiten" wird das Dialogfeld "Zusätzliche extrahierte Daten" geöffnet.

Geben Sie die Werte für die folgenden Optionen ein:

XPath-Ausdruck

Gibt den XPath-Ausdruck an, der unter "Zusätzliche extrahierte Daten" ausgewählt wurde.

Datensatzvariable

Gibt den Namen einer Operatordatensatzvariablen an, in der Werte gespeichert werden sollen, die anhand des ausgewählten XPath-Ausdrucks extrahiert wurden.

Typ

Gibt den Typ der Elemente an, die aus der Antwort extrahiert werden. Wählen Sie einen der folgenden gegenwärtig unterstützten Typen aus:

- Ganzzahl
- Array mit ganzen Zahlen
- Zeichenfolge
- Zeichenfolgen-Array
- XML-Fragment
- XML-Fragment-Array

Ausgabeparameter

inputSource

ExtractedVarInfo (ValueMap)

xPathQuery

dataSetVa

type

expressionVal

xmlInputFileName

isStripXMLNamespaces

Anwenden von XSLT-Operatoren

XSLT übernimmt vordefinierte Formatvorlagen, um ein XML-Quelldokument in ein anderes darstellungsbezogenes Format wie HTML, XHTML oder SVG zu transformieren.

Eingabeparameter

XML-Eingabequelle

Definiert das Quell-XML-Dokument, das in eines der folgenden Formate transformiert werden soll:

Ausdruck

Definiert ein Muster für das Identifizieren einer Zeichenfolge von Werten. Zum Beispiel können Sie den Ausdruck *Datasets["/VER2_Dataset"].srcXML* definieren

Pfad der XML-Datei

Definiert den Pfad einer Datei, in der ein Operator ausgeführt wird. Der Dateipfad kann ein freigegebener Speicherort oder eine URL sein.

Lesen Sie sich folgende Beispiele sorgfältig durch:

- Dateipfad: *C:\\XSLTV2\\books.xml*
- Freigegebener Speicherort: *\\\\user-xp1\\user\\XSLTV2\\books.xml*
- URL: *http://<serverName>:CA Portal/c2orepository/XSLTV2/books.xml*

Inline-XML

Gibt die XML-Daten an, die als Eingabe dienen.

XSLT-Eingabequelle

Gibt die Quell-XSLT-Informationen in eines der folgenden Formate an:

Ausdruck

Definiert ein Muster für das Identifizieren einer Zeichenfolge von Werten. Zum Beispiel können Sie den Ausdruck *Datasets["/VER2_Dataset"].srcXML* definieren

Pfad der XSL-Datei

Definiert den Pfad einer Datei, in der ein Operator ausgeführt wird. Der Dateipfad kann ein freigegebener Speicherort oder eine URL sein.

Lesen Sie sich folgende Beispiele sorgfältig durch:

- Dateipfad: *C:\\XSLTV2\\books.xsl*
- Freigegebener Speicherort: *\\\\user-xp1\\user\\XSLTV2\\books.xsl*
- URL: *http://<serverName>:CA Portal/c2orepository/XSLTV2/books.xsl*

Inline-XSL

Gibt die XSL-Daten an, die als Eingabe dienen.

XSLT Version (Produktversion)

Gibt eine der folgenden Optionen an, um die XSLT-Version zu ermitteln, die verwendet wird, um das Quell-XML zu transformieren:

- Version 1
- Version 2
- Festgelegt in XSLT

Eingabeparameter

Gibt die Eingabeparameter als Schlüssel-Wert-Paare im XSLT-Operator an. Sie können den Eingabeparametern (Schlüssel), die in XSL definiert sind, Werte dynamisch zuweisen.

Sie können einen Schlüssel (top-author) in XSL definieren und einen Wert (Robert Kisosk) von einem XSLT-Operator zuweisen, wie im folgenden Beispiel dargestellt:

```
<xsl:param name="top-author">Jasper Forde</xsl:param>

Key: top-author Value="Robert Kisosk"
```

Ausgabeparameter

Zeigt die vordefinierten Ausgabeparameter als Schlüssel-Wert-Paare an. Die Ausgabeparameter sind XSLT-versionsspezifisch und basieren auf XSLT-Standards. Weitere Informationen zu Ausgabeparameter finden Sie unter http://www.w3.org/TR/xslt#output.

XML-Ausgabe

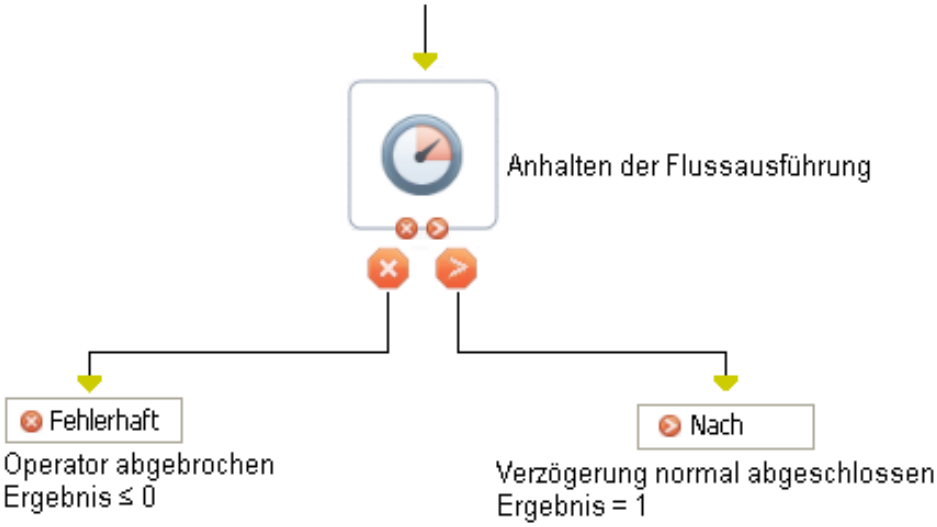
Sie können die XML-Ausgabe in der Datensatzvariable oder in einer im Pfad der Ausgabedatei platzierten Datei anzeigen.

Sie können einen Wert (yes) dem Schlüssel (include-content-type) von einem XSLT-Operator zuweisen, wie im folgenden Beispiel dargestellt:

include-content-type="yes"

Operator "Verzögerung"

Der Operator "Verzögerung" verzögert die Verarbeitung von nachfolgenden Verzweigungen eines Prozesses bis ein angegebener Zeitraum von Tagen, Stunden, Minuten und Sekunden verstrichen ist. Die Verzögerung kann sich auf den Start der Verarbeitung des Operators "Verzögerung" oder den Start der Verarbeitung des Prozesses beziehen.



Eingabeparameter

Tage

Gibt die Anzahl von Tagen an, um die die Verarbeitung nachfolgender Verzweigungen des Prozesses verzögert werden soll.

Stunden

Gibt zusätzliche Stunden an, um die die Verarbeitung von nachfolgenden Verzweigungen des Prozesses verzögert wird.

Um den Teil eines Tages in Stunden anzugeben, geben Sie 0 bis 23 ein.

Hinweis: Die Anzahl von Stunden ist ein Ausdruck, daher gibt es keine Validierungen.

Minuten

Gibt zusätzliche Minuten an, um die die Verarbeitung von nachfolgenden Verzweigungen des Prozesses verzögert wird.

Um den Teil einer Stunde in Minuten anzugeben, geben Sie 0 bis 59 ein.

Hinweis: Die Anzahl von Minuten ist ein Ausdruck, daher gibt es keine Validierungen.

Sekunden

Gibt zusätzliche Sekunden an, um die die Verarbeitung von nachfolgenden Verzweigungen des Prozesses verzögert wird.

Um den Teil einer Minute in Sekunden anzugeben, geben Sie 0 bis 59 ein.

Hinweis: Die Anzahl von Sekunden ist ein Ausdruck, daher gibt es keine Validierungen.

Auf Prozess-Anfangszeit bezogen

Sorgt dafür, dass sich die Verzögerung auf den Start des Prozesses bezieht.

Wenn dieses Kontrollkästchen deaktiviert wird, beginnt die Verzögerung nachdem der Prozess mit der Verarbeitung des Operators begonnen hat. Zum Beispiel kann diese Option verwendet werden, um einen Alarm auszulösen, wenn der Prozess nicht innerhalb eines angegebenen Zeitraums beendet wird (den Operator "Ende" erreicht).

Ausgabeparameter

flowchart_start_time

Tage

Stunden

Minuten

Sekunden

Relative_to_Flowchart

TargetTime

targetDate

Operator "Java aufrufen"

Der Operator "Java aufrufen" nutzt die Funktionen optimal aus, die in den externen JAR-Dateien (oder .class-Dateien) in CA Process Automation enthalten sind. Sie identifizieren die JAR-Dateien oder die .class-Dateien, indem Sie den Speicherort in den Operator-Eingabeparametern angeben.

Sobald die JAR-Datei gefunden wird, können Sie Java-Code schreiben, der sich auf die Klassen in der JAR-Datei bezieht. Sie können Variablen an diesen Code übergeben, indem Sie die Eingabeparameter des Operators "Java aufrufen" verwenden. Der Code, den Sie schreiben, geht in der Hauptmethode des Operators.

Sie können angeben, ein Java-Objekt im Operatordatensatz zu speichern, nachdem der Code durch den Operator ausgeführt wird. Das Java-Objekt wird in CA Process Automation unter dem Datentyp "JavaObject" gespeichert. Sie können diesen JavaObject-Datentyp für nachfolgende "Java aufrufen"-Operatoren verfügbar machen.

Angenommen beispielsweise, Sie möchten den Operator "Java aufrufen" in einem Prozess verwenden. Sie schließen ihn in einen Prozess im Prozess-Editor ein und nennen ihn "Java-Operator 1". Sobald der Operator ausgeführt wird, wird das Java-Objekt im Operatordatensatz gespeichert und als JavaObject-Datentyp angezeigt. Jetzt verfügen Sie über einen weiteren Operator "Java aufrufen" in Ihrem Prozess. Sie können das gleiche JavaObject des Java-Operator 1 in Ihrem neuen Java-Operator 2 verwenden. Sie übergeben das gespeicherte Objekt von Java-Operator 1 an Java-Operator 2 als JavaObject-Datentyp.

Der Operator "Java aufrufen" kann nicht auf einem Koordinationsrechner ausgeführt werden. Dieser Operator kann nur auf einem Agenten ausgeführt werden.

Eingabeparameter

Nachfolgend sind die Eingabeparameter für den Operator "Java aufrufen" aufgeführt.

Code

Externe JAR-Dateipfade auflisten?

Wählen Sie diese Option aus, um eine Liste der Pfade zu den erforderlichen externen JAR-Dateien im Feld "Externe JAR-Dateipfade als Ausdruck" anzugeben.

Deaktivieren Sie diese Option, um die Pfade zu den erforderlichen externen JAR-Dateien als Ausdruck anzugeben, d. h. als Datensatzvariable im Feld "Externe JAR-Dateipfade als Ausdruck".

Dieses Kontrollkästchen ist standardmäßig aktiviert.

Externe JAR-Dateipfade

Gibt die Liste der Pfade für die externen JAR-Dateien an, die vom Code der Hauptmethode des Operators (sofern vorhanden) benötigt wird.

Der Operator "Java aufrufen" lädt die JAR-Dateien, die in diesem Feld aufgelistet sind. Alle in diese Liste eingegebenen JAR-Dateien sind für den Java-Code verfügbar, der von diesem Operator ausgeführt wird. Die Klassen, die in den JAR-Dateien auf Operatorebene definiert sind, überschreiben die gleichen Klassen, die in den JAR-Dateien auf Ebene der Operatorkategorie angegeben sind.

Für jeden Pfad gibt es folgende Möglichkeiten:

- Geben Sie den vollständigen Pfad zu einer JAR-Datei ein, die sich auf dem Host befindet, auf dem der (zum Kontaktpunkt zugeordnete) CA Process Automation-Agent ausgeführt wird. Der vollständige Pfad wird folgendermaßen angegeben:
  - Beginnt mit: /
  - Beginnt mit: \\
  - Aus dem Formular: ^.:.* (ein regulärer Ausdruck, der mit einem Zeichen beginnt, gefolgt von einem Doppelpunkt - : - und dem Rest der Zeichenfolge.)
- Geben Sie den Pfad zu einer JAR-Datei ein, die über HTTP heruntergeladen werden kann. Stellen Sie sicher, dass der HTTP-Pfad keine Authentifizierung benötigt und keinen HTTP-Proxy verwendet. Der Pfad zu den JAR-Dateien muss mit http:// oder https:// beginnen.

- Es wird davon ausgegangen, dass ein anderer Pfad ein relativer Pfad zu einer JAR-Datei ist, die in den Anwenderressourcen von CA Process Automation hochgeladen wurde. CA Process Automation hängt den Pfad der JAR-Datei an den Pfad des Verzeichnisses "CA Process Automation-Anwenderressourcen" des Agenten (der dem Kontaktpunkt zugeordnet ist), der den Operator ausführt.

  Starten Sie den relativen Pfad der JAR-Datei nicht mit:

  - /
  - \\

  Anderenfalls nimmt CA Process Automation an, dass der Pfad der JAR-Datei ein vollständiger Pfad ist.

Ressourcen innerhalb von CA Process Automation, einschließlich der Anwenderressourcen, werden innerhalb des Spiegelungsintervalls des Agenten gespiegelt. Stellen Sie sicher, dass die JAR-Dateien, die in den Anwenderressourcen hochgeladen wurden, bereits gespiegelt sind, bevor sie im Operator "Java aufrufen" verwendet werden.

class-Dateien

Zusätzlich zu externen JAR-Dateien, können Sie folgendermaßen .class-Dateien laden:

- Bei .class-Dateien in einem umbenannten Paket geben Sie einen Pfad ein, der mit dem Verzeichnis endet, das die .class-Dateien enthält.

  Zum Beispiel wenn "MyAccount.java" zu keinem Paket gehört und "MyAccount.class" sich hier befindet:

  C:\java\tests\MyAccount.class

  ...dann geben Sie an, dass der Operator folgenden Pfad verwenden soll:

  C:\\java\\tests

- Bei .class-Dateien in einem benannten Paket geben Sie einen Pfad ein, der mit dem Verzeichnis endet, in dem das "root"-Paket enthalten ist. Dieses Paket ist im vollständigen Paketnamen auch als erstes Paket bekannt.

  Zum Beispiel wenn "MyAccount.java" zu dem Paket "com.ca.tech" gehört, und "MyAccount .class" sich unter "C:\java\othertests\com\ca\tech\MyAccount.class" befindet

  ...dann geben Sie an, dass der Operator folgenden Pfad verwenden soll:

  C:\\java\\othertests

Wenn Sie den Pfad in ein Verzeichnis angeben (um .class-Dateien zu laden), geben Sie ihn als vollständigen Pfad ein. Sie können ihn auch als relativen Pfad zu CA Process Automation-Anwenderressourcen eingeben. Geben Sie den Pfad nicht als HTTP-Pfad ein.

Geben Sie den Pfad in ein Verzeichnis an, um .class-Dateien und keine JAR-Dateien zu laden. Im Gegensatz zu .class-Dateien, benötigt jede JAR-Datei einen separaten Pfad, der mit der JAR-Datei endet (nicht mit dem Verzeichnis, in dem sie sich befindet).

Externe JAR-Dateipfade als Ausdruck

Gibt die Datensatzvariable der indizierten Zeichenfolgen an, die die Liste der Pfade zu den externen JAR-Dateien enthält, die der Code der Hauptmethode des Operators benötigt.

Erforderliche Hauptmethode eingeben?

Wählen Sie diese Option aus, um den Java-Code im Feld "Erforderliche Hauptmethode" anzugeben.

Deaktivieren Sie diese Option, um den Java-Code als eine Datensatzvariable im Feld "Erforderliche Hauptmethode als Ausdruck" anzugeben.

Dieses Kontrollkästchen ist standardmäßig aktiviert.

Erforderliche Hauptmethode

Gibt den Java-Codetext der Hauptmethode an. Sie können nach allen Dateien suchen, die diesen Code enthalten. Die Hauptmethode besteht aus normalen Java-Anweisungen und Ausdrücken. Sie müssen Objekte initialisieren und verwenden, die im Java-SDK oder in externen JAR-Dateien definiert sind.

Hinweis: Wenn Sie auf "OK" klicken, analysiert CA Process Automation den Code und überprüft die strukturelle Validierung. Eine Fehlermeldung wird angezeigt, wenn ein Fehler in der Struktur des Codes gefunden wird.

Weitere Informationen finden Sie unter Java-Code im Operator "Java aufrufen" (siehe Seite 424).

Erforderliche Hauptmethode als Ausdruck

Gibt die Datensatzvariable an, die den Inhalt der Hauptmethode enthält. Die Hauptmethode besteht aus normalen Java-Anweisungen und Ausdrücken. Sie müssen Objekte initialisieren und verwenden, die im Java-SDK oder in externen JAR-Dateien definiert sind. Es erfolgt keine strukturelle Validierung.

Weitere Informationen finden Sie unter Java-Code im Operator "Java aufrufen" (siehe Seite 424).

Kontext-Klassenladeprogramm festlegen?

Legen Sie dieses Feld abweichend vom Standardwert fest, wenn Ihre Hauptmethode bzw. Ihre externe JAR-Datei zum Laden das Java-Kontext-Klassenladeprogramm benötigt. Legen Sie als Java-Kontext-Klassenladeprogramm entweder das Operator-Klassenladeprogramm oder das Modul-Klassenladeprogramm fest, um eine Ausnahme des Typs "ClassCastException" zu vermeiden.

Der Operator "Java aufrufen" verwendet eine Kette von Klassenladeprogrammen, um Klassen zu laden, während der Java-Code ausgeführt wird. Diese Kette ist folgendermaßen aufgebaut (neben weiteren Klassenladeprogrammen):

1. Operator-Klassenladeprogramm: Ein Klassenladeprogramm für das laden der Klassen, die auf Operatorebene angegeben wurden.
2. Modul-Klassenladeprogramm: Ein Klassenladeprogramm für das laden der Klassen, die auf Modulebene angegeben wurden.
3. Kontext-Klassenladeprogramm
4. Normales Java Class.ForName

Der Operator konsultiert die angegebenen Klassenladeprogramme nacheinander und wechselt zum nächsten, wenn eine Klasse nicht gefunden wird.

Diese Kette funktioniert, solange der Code, den Sie ausführen, nicht ausdrücklich sein eigenes Klassenladeprogramm verwendet. In diesem Fall wird in den Protokollen "ClassCastException" angezeigt.

Beispiel:

Beachten Sie, dass die Klasse "MyChildXMLParser" eine Erweiterung der Klasse "MyParentXMLParser" ist. Das folgende Codefragment erstellt eine MyParentXMLParser-Klasse durch die Verwendung einer Java-Factory. Diese Factory lädt und erstellt eine MyChildXMLParser-Klasse, die in ein MyParentXMLParser-Objekt umgeformt werden kann:

```
public MyParentXMLParser() {
        super((MyParentXMLParser)ObjectFactory.createObject("co
m.ca.parser.MyChildXMLParser"));
          ....
     }
```

Beachten Sie in diesem Beispiel Folgendes:

- Die Methode "ObjectFactory.createObject()" ruft ihr eigenes Klassenladeprogramm auf, um die MyChildXMLParser-Klasse zu laden. Dieses Verhalten ist bei Java-Factorys üblich. Sie verwenden entweder das System-Kassenladerprogramm oder, wenn vorhanden, ein Kontext-Klassenladeprogramm, um die Klasse zu laden (anstelle des Klassenladeprogramms, das vom ausgeführten Programm verwendet wird).
- Die MyParentXMLParser-Klasse wird vom Klassenladeprogramm des Operators "Java aufrufen" geladen (entsprechend der zuvor ausgewählten Kette).
- ((MyParentXMLParser)ObjectFactory.createObject) führt zu einer Ausnahme des Typs "ClassCastException". Obwohl MyChildXMLParser eine Erweiterung von MyParentXMLParser ist, wurden die beiden Klassen von unterschiedlichen Klassenladeprogrammen geladen. Dadurch sind sie komplett unterschiedlich.

Um dieses Problem zu lösen, legen Sie im Feld "Kontext-Klassenladeprogramm festlegen" einen der folgenden Werte fest:

- "1 : Operator-Klassenladeprogramm": Wenn die JAR-Datei, die die Klassen "MyChildXMLParser" und "MyParentXMLParser" enthält, in den Operatoreigenschaften angegeben ist.
- "2 : Modul-Klassenladeprogramm": Wenn die JAR-Datei, die die Klassen "MyChildXMLParser" und "MyParentXMLParser" enthält, in den Moduleigenschaften angegeben ist.

Hinweis: "0 : Standard" wird in allen anderen Fällen verwendet, in denen Ihr Java-Code nicht ausdrücklich Klassen mithilfe des eigenen Klassenladeprogramms lädt. Dieser Wert ist der Standardwert dieses Felds.

Wenn das Kontext-Klassenladeprogramm auf das Klassenladeprogramm des Operators "Java aufrufen" festgelegt wird, wird erzwungen, dass die Java-Factorys, die vom Code des Anwenders angerufen werden, das Klassenladeprogramm des Operators "Java aufrufen" verwenden. Diese Aktion entfernt "ClassCastException".

Java-Code im Operator "Java aufrufen"

Wenn Sie den Operator "Java aufrufen" verwenden, beachten Sie die folgenden Richtlinien zur Implementierung von Java-Code:

- CA Process Automation führt den Java-Code in einem BeanShell-Interpreter aus. Verwenden Sie in folgenden Fällen diesen Operator mit Java-Code-Syntax oder BeanShell-Skriptsyntax *ohne* die Verwendung von BeanShell-Befehlen:
  - BeanShell-Befehle funktionieren nicht unter "Strict Java Mode" (auf der Modulebene festgelegt).
  - BeanShell-Befehle, die "classpath" ändern, werden nicht empfohlen. Sie können Auswirkungen darauf haben, wie CA Process Automation Java-Objektinstanzen aus dem ausgeführten Java-Code in den Datensatz des Operators speichert.
  - BeanShell-Befehle, die "classpath" ändern, können Auswirkungen darauf haben, wie CA Process Automation Java-Objektinstanzen aus einem CA Process Automation-Datensatz in den Code lädt.

  Weitere Informationen zur BeanShell-Syntax und zu Befehle finden Sie unter: http://www.beanshell.org/
- Sie können die standardmäßigen Modifikatoren der Java-Variable auf eingegebene Variablen verwenden:
  - private / protected / public
  - transient
  - volatile
  - static
  - final

    Der BeanShell-Interpreter implementiert nur "final" (und ignoriert die anderen).
- Sie können die standardmäßigen Java-Modifikatoren für Methoden verwenden:
  - private / protected / public
  - final
  - native
  - abstract
  - static
  - synchronized

    Nur "synchronized" ist derzeit implementiert. Der BeanShell-Interpreter ignoriert die anderen.

- Schließen Sie alle Klassendefinitionen in externen JAR-Dateien ab, und verwenden Sie sie im Code der Hauptmethode des Operators.
- "Java rt.jar" enthält alle Java-Kernbibliotheken und wird zur Laufzeit automatisch in den "classpath" des Operators eingefügt.
- Das von CA Process Automation verwendeten JAR-Dateien befinden sich zur Laufzeit im "classpath" des Operators. Ihr Code kann funktionieren, auch ohne Auflistung aller benötigten JAR-Dateien im Operator/in der Kategorie, wenn Sie die Klassen verwenden, die bereits von CA Process Automation verwendet werden.
- Allgemeine Java-Kernpakete und einige Erweiterungen werden zur Laufzeit automatisch in Ihren Java-Code importiert. Es ist nicht notwendig, dass Sie sie in Ihren Code importieren. Zu den Paketen gehören:
  - javax.swing.event
  - javax.swing
  - java.awt.event
  - java.awt
  - java.net
  - java.util
  - java.io
  - java.lang
  - bsh.EvalError
  - bsh.Interpreter

- Der Java-Code kann aus normalen Java-Anweisungen und Ausdrücken bestehen. Sie können auch Ihre eigenen Methoden definieren und sie im Code verwenden. Es folgt ein Beispiel:

```
// Import the classes that you want to use
import ca.tech.pam.MyAccount;
// Note: no need to import StringBuffer and Date because they are part of the
// automatically imported packages
// import java.lang.StringBuffer;
// import java.util.Date;
// Note: the jar that contains the ca.tech.pam.MyAccount class
// must be in the list of External Jars of the operator or the module;
// but java lang and java util are in rt.jar, which is automatically put in the classpath
MyAccount acct = new MyAccount(newDate(),100);
// Use the public methods of the MyAccount object
acct.addFunds(34);
acct.subFunds(10);
// Define your own method
String getStatement(MyAccount acc) {
    StringBuffer strBuff = new StringBuffer("Account Balance: " + acc.getBalance());
    Date dt = new Date(System.currentTimeMillis());
    strBuff.append(" on date: " + dt);
    return strBuff.toString();
}
// Use the method you defined
// also print the statement using the 'logger' object that you
// setup in the 'Logger' page of the operator
logger.info(getStatement(acct));
```

Hinweis: Um diese Anweisung auszuführen, legen Sie die Protokollierung auf "Wahr" fest, um den Namen der Protokolldatei anzugeben. Anderenfalls tritt während der Ausführung ein Fehler auf.

Am Ende der Ausführung enthält die Protokollmeldung:

```
Account Balance: 124.  on date: Wed Jul 13 12:53:37 EDT 2011
```

(Die Meldung enthält das richtige Datum und die Uhrzeit der Ausführung.)

Eingabe/Ausgabe

Parameter

Die CA Process Automation-Parameter, die an die Hauptmethode übergeben werden sollen. Geben Sie die Parameter in derselben Reihenfolge ein, in der sie an die Hauptmethode weitergegeben werden. Lassen Sie dieses Feld leer, wenn keine Parameter erforderlich sind.

Nur einfache CA Process Automation-Parametertypen können folgendermaßen an die Hauptmethode übergeben werden:

- PAM-Boolean wird als Boolean-Objekt übergeben.
- PAM-Date wird als Date-Objekt übergeben.
- PAM-Double wird als Double-Objekt übergeben.
- PAM-Integer wird als Integer-Objekt übergeben.
- PAM-Long wird als Long-Objekt übergeben.
- PAM-String wird als String-Objekt übergeben.
- PAM-Object Reference wird als String-Objekt übergeben.
- PAM-JavaObject wird deserialisiert und in eine Java-Objektinstanz des ursprünglichen Klassentyps geladen und wird anschließend an den Java-Code übergeben.

Hinweis: Der Operator (oder die Operatorkategorie) muss den Pfad zu der JAR-Datei enthalten, die die Klassendefinition dieses Objekts enthält. Anderenfalls schlägt der Operator mit folgender Begründung fehl:

Class Not Found Error when deserializing object. Make sure the class jar is in the operator or module list of jars.

Komplexe CA Process Automation-Parametertypen (indizierte Typen, ValueMaps usw.) können *nicht* an den Java-Code übergeben werden.

Die Hauptmethode kann auf die übergebenen Parameter über das args-Array der Objekte zugreifen:

- args[0] entspricht dem ersten Parameter in der Liste.
- args[1] entspricht dem zweiten Parameter in der Liste usw.

Ausgabevariablen-Namen

Die Namen der Variablen, die im Operatordatensatz gespeichert werden, nachdem die Ausführung der Hauptmethode abgeschlossen wurde. Diese Variablen müssen im Bereich der Hauptmethode definiert werden. Lassen Sie dieses Feld leer, wenn keine Ausgabevariablen im Operatordatensatz gespeichert werden sollen.

Die Ausgabevariablen werden folgendermaßen gespeichert:

- Boolean-Objekt wird als PAM-Boolean gespeichert
- Date-Objekt wird als PAM-Date gespeichert

- Integer-Objekt wird als PAM-Integer gespeichert
- Number-Objekt wird als "PAM-Long"- oder "PAM Double"-Objekt gespeichert
- String-Objekt wird als PAM-String gespeichert
- Character-Objekt wird als PAM-String gespeichert
- Ein Array von einem dieser aufgelisteten Objekte wird als ein indizierter CA Process Automation-Typ gespeichert. Der Typ des ersten Objekts im Array der Objekte definiert den CA Process Automation-Typ.
- "Undefined" wird als CA Process Automation-Zeichenfolge mit dem Wert "undefined" gespeichert.
- Ein anderes Java-Objekt, das hier nicht aufgelistet ist, wird serialisiert und als ein CA Process Automation-JavaObject gespeichert.

  Hinweis: Das Java-Objekt muss serialisiert (implementiert "java.io.Serializable") werden können, um als CA Process Automation-JavaObject gespeichert werden zu können. Anderenfalls schlägt der Operator mit folgender Begründung fehl:

  `Fehler wenn ein Objekt der Klasse x serialisiert wird. Objekt kann nicht serialisiert werden.`

  Wobei *x* der Klassenname des Objekts ist.

Protokollierung

Protokollierung verwenden?

Legen Sie dieses Feld auf "Wahr" fest, um eine Instanz des Objekts "org.apache.log4j.Logger" auf Daten in die angegebene Protokolldatei zu schreiben. Die Protokollierung bearbeitet das Öffnen und Schließen der Datei. Die Protokollierung ist im Kontext der Hauptmethode verfügbar und kann als 'logger.debug()', 'logger.info()' usw. verwendet werden.

Wahr

Fordert den Operator auf, den Wert "Wahr" zu verwenden. Der Operator verwendet eine Instanz von "Protokollierung".

False

Fordert den Operator auf, den Wert "Falsch" zu verwenden. Der Operator verwendet keine Instanz von "Protokollierung".

Leer

Fordert den Operator auf, den Wert zu verwenden, der im Feld "Standardmäßige Protokollierung verwenden" der Operatorkategorie festgelegt ist. Wenn dieser Wert auf Ebene der Operatorkategorie leer ist, wird "Protokollierung verwenden?" standardmäßig auf "Falsch" gesetzt.

Ein anderer Wert fordert den Operator auf, den Wert "Falsch" zu verwenden, und der Operator verwendet keine Instanz von "Protokollierung".

Wenn eine Instanz von "Protokollierung" verwendet wird, dann wird sie im Zusammenhang der Hauptmethode des Operators verfügbar sein. "Protokollierung" wird folgendermaßen verwendet:

- logger.debug("my log message")
- logger.info("my log message")
- logger.warn("my log message")
- logger.error("my log message")
- logger.fatal("my log message")

Wenn eine Instanz von "Protokollierung" nicht verwendet wird, ist das Objekt "Protokollierung" im Zusammenhang der Hauptmethode des Operators nicht vorhanden.

Pfad der Protokolldatei

Der Pfad zur vom Code verwendeten Protokollierung. Dieser Pfad muss auf eine Datei verweisen, die sich auf dem CA Process Automation-Agent-Host befindet. Wenn dieses Feld leer ist, erbt der Operator den Wert, der im Feld "Standardmäßiger Pfad der Protokolldatei" der Operatorkategorie festgelegt ist.

Protokollebene

Geben Sie die Protokollebene der Protokollierung an.

0

Fordert den Operator auf, "DEBUG" zu verwenden, was dazu führt, dass die Protokollierung die Protokollmeldungen "Debug", "Info", "Warn", "Error" und "Fatal" schreibt.

1

Fordert den Operator auf, "INFO" zu verwenden, was dazu führt, dass die Protokollierung die Protokollmeldungen "Info", "Warn", "Error" und "Fatal" schreibt.

2

Fordert den Operator auf, "WARN" zu verwenden, was dazu führt, dass die Protokollierung die Protokollmeldungen "Warn", "Error" und "Fatal" schreibt.

3

Fordert den Operator auf, "ERROR" zu verwenden, was dazu führt, dass die Protokollierung die Protokollmeldungen "Error" und "Fatal" schreibt.

4

Fordert den Operator auf, "FATAL" zu verwenden, was dazu führt, dass die Protokollierung die Protokollmeldungen "Fatal" schreibt.

Leer

Fordert den Operator auf, den Wert zu erben, der im Feld "Standardmäßige Protokollebene" der Operatorkategorie festgelegt ist. Wenn dieser Wert auf Ebene der Operatorkategorie leer ist, wird "Protokollebene" standardmäßig auf "Debug" festgelegt.

Ein anderer Ganzzahlwert

Fordert den Operator auf, den Wert "DEBUG" zu verwenden.

Hinweis: Sie können die Protokollebene zur Laufzeit in der Hauptmethode des Operators überschreiben. Dieses Beispiel legt die Protokollebene auf "Fatal" fest:

```
import org.apache.log4j.Level;
logger.setLevel((Level) Level.FATAL);
```

An Protokolldatei anhängen?

Legen Sie dieses Feld auf "Wahr" fest, um Daten von diesem Operator an die Protokolldatei (wenn vorhanden) anzuhängen.

Wahr

Fordert den Operator auf, den Wert "Wahr" zu verwenden. Der Operator wird an die Protokolldatei angehängt.

False

Fordert den Operator auf, den Wert "Falsch" zu verwenden. Der Operator löscht den Inhalt der vorhandenen Protokolldatei, bevor die neuen Daten aus dem Operator geschrieben werden.

Leer

Fordert den Operator auf, den Wert zu verwenden, der im Feld "An standardmäßige Protokolldatei anhängen?" der Operatorkategorie festgelegt ist. Je nach dem wie der Wert festgelegt ist, kann der Operator zur Protokolldatei angehängt werden. Wenn dieser Wert auch auf Ebene der Operatorkategorie leer ist, wird "An Protokolldatei anhängen?" standardmäßig auf "Falsch" gesetzt.

Ein anderer Wert fordert den Operator auf, den Wert "Falsch" zu verwenden, und der Operator wird nicht zur Protokolldatei angehängt.

Daten ohne Protokollierungsinformationen ins Protokoll schreiben?

Setzen Sie diesen Wert auf "Wahr", sodass die Daten ohne zusätzliche Protokollierungsinformationen geschrieben werden. Nur die Protokollmeldung wird geschrieben.

Setzen Sie diesen Wert auf "Falsch", um zusätzliche Protokollierungsinformationen im folgenden Format zu schreiben:

```
Day Month Year Hours:Minutes:Secs Log_level [UUID of the Invoke_Java operator that logged this message]: log message
```

Wahr

Fordert den Operator auf, den Wert "Wahr" zu verwenden. Die Protokollierung schreibt Daten ohne zusätzliche Protokollierungsinformationen.

False

Fordert den Operator auf, den Wert "Falsch" zu verwenden. Die Protokollierung schreibt Daten mit zusätzlichen Protokollierungsinformationen.

Leer

Fordert den Operator auf, den Wert zu verwenden, der im Feld "Daten standardmäßig ohne Protokollierungsinformationen ins Protokoll schreiben?" der Operatorkategorie festgelegt ist. Wenn dieser Wert auf Ebene der Operatorkategorie leer ist, wird "Daten ohne Protokollierungsinformationen ins Protokoll schreiben?" standardmäßig auf "Falsch" gesetzt.

Ein anderer Wert fordert den Operator auf, den Wert "Falsch" zu verwenden. Die Protokollierung schreibt Daten mit zusätzlichen Protokollierungsinformationen.

Ausgabeparameter

ErrorLineNumber

Wenn ein Fehler aufgrund der ausgeführten Hauptmethode auftritt, enthält diese Variable die Nummer der Codezeile, die einen Fehler verursacht (sofern vorhanden). Dieses Feld ist leer, wenn aufgrund des ausgeführten Java-Codes kein Fehler auftritt.

ErrorMessage

Wenn ein Fehler aufgrund der ausgeführten Hauptmethode auftritt, enthält diese Variable die Fehlermeldung. Dieses Feld ist leer, wenn aufgrund der ausgeführten Hauptmethode kein Fehler auftritt.

ErrorRoot

Wenn ein Fehler aufgrund der ausgeführten Hauptmethode auftritt, enthält diese Variable die Codezeile, die einen Fehler verursacht (sofern vorhanden). Dieses Feld ist leer, wenn aufgrund der ausgeführten Hauptmethode kein Fehler auftritt.

ErrorException

Wenn die ausgeführte Hauptmethode eine Ausnahme ausgibt, enthält diese Variable die ausgegebene Ausnahme. Dieses Feld ist leer, wenn aufgrund der ausgeführten Hauptmethode kein Fehler auftritt.

Ergebnis

1

Der Operator wird erfolgreich abgeschlossen.

-1

Der Operator schläft fehl.

Ursache

Abgeschlossen

Der Operator wird erfolgreich abgeschlossen.

Fehlermeldung

Der Operator schläft fehl. Wenn ein Fehler aufgrund der ausgeführten Hauptmethode auftritt, enthält diese Variable "ErrorMessage", "ErrorLineNumber", "ErrorRoot" und "ErrorException" (sofern zutreffend).

ResponseMessage

ResponseCode

externalOpJarsType

externalOpJars

externalOpJarsExpression

inlineScriptType

inLineScript

scriptExpression

parameters

outputVariables

useLogger

logFile

logLevel

appendToLogFile

useSimpleLoggerLayout

Java-Beispiel

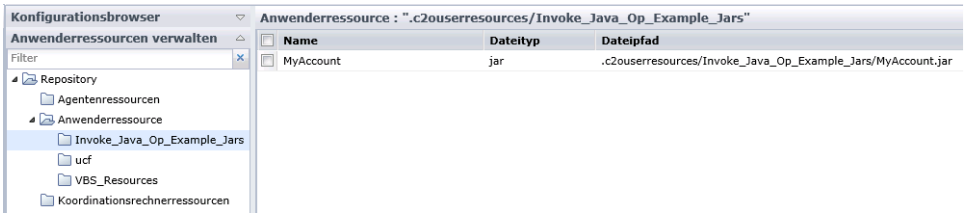
Folgendes Beispiel befindet sich im Beispielabschnitt des Feldes "Erforderliche Hauptmethode".

```
/*
Die Hauptmethode wird verwendet, um Objekte und Methoden aufzurufen, die im Java-SDK oder in externen
JAR-Dateien definiert sind.
Die Hauptmethode besteht aus normalen Java-Anweisungen und Ausdrücken.
Sie können auch:
- Definieren Sie Ihre eigenen Methoden, und verwenden Sie sie in der Hauptmethode
- Übergeben Sie Eingabeparameter an die Hauptmethode
- Speichern Sie Ausgabevariablen im Datensatz des Operators am Ende der Ausführung der Hauptmethode
- Verwenden Sie ein Protokollierungsobjekt in der Hauptmethode

Normalerweise würden Sie alle Klassendefinitionen in externen JAR-Dateien abschließen und sie in den
externen JAR-Dateipfaden des Operators auflisten, dann initialisieren und diese Objekte in der
Hauptmethode verwenden.
*/

/*
Das untere Beispiel zeigt, wie ein MyAccount-Objekt initialisiert und verwendet wird, das in einer
externen JAR-Datei definiert ist.
Operatorkonfiguration:
1. Geben Sie den Pfad an:
"Invoke_Java_Op_Example_Jars/MyAccount.jar" in der Liste des Operators der externen JAR-Dateipfade.
"MyAccount.jar" enthält die Klasse "ca.tech.pam.MyAccount", die im unteren Code verwendet wird.
Während der Installation von CA Process Automation wird "MyAccount.jar" als Anwenderressource
hochgeladen.
2. Geben Sie eine Datensatzvariable des Typs "Datum" als erstes Objekt in der Liste der Eingabeparameter
des Operators an.
    Auf diesen Parameter kann in der Hauptmethode als "args[0]" zugegriffen werden.
3. Geben Sie eine Datensatzvariable des Typs "Ganzzahl" und den Wert 100 als zweites Objekt in der
Liste der Eingabeparameter des Operators an.
    Auf diesen Parameter kann in der Hauptmethode als "args[1]" zugegriffen werden.
4. Geben Sie die Variablennamen "acct" (ohne Anführungszeichen) als erstes Objekt in der Liste der
Ausgabevariablen-Namen des Operators an.
    "acct" wird in der Hauptmethode als "MyAccount"-Objekt erstellt, sodass am Ende der Ausführung
des Operators "acct" im Datensatz des Operators als Variable des Typs "JavaObject" gespeichert wird.
5. Konfigurieren Sie den Operator für die Verwendung einer Protokollierung, und legen Sie "Pfad der
Protokolldatei" auf einen lokalen Dateipfad, "Protokollebene" auf "Info", "An Protokolldatei
anhängen?" auf "falsch" und "Daten ohne Protokollierungsinformationen ins Protokoll schreiben?" auf
"wahr" fest.
*/

// Importieren Sie die Klassen, die Sie verwenden wollen
import ca.tech.pam.MyAccount;

// Hinweis\: StringBuffer und Datum (unten verwendet) brauchen nicht importiert zu werden, da sie Teil
der
```

```
// automatisch importierten Pakete sind. (Die vollständige Liste dieser Pakete ist in der Dokumentation
enthalten.)
// import java.lang.StringBuffer;
// import java.util.Date;

// Initialisieren Sie das Objekt 'MyAccount'
// Beachten Sie, dass der MyAccount-Konstruktor in der externen JAR-Datei definiert ist als\:
// public MyAccount(Date date, int balance)
MyAccount acct = new MyAccount(args[0], args[1]);

// Verwenden Sie die öffentlichen Methoden des Objekts 'MyAccount'.
// Note that addFunds is defined in the external jar as:
// public int addFunds (int amnt)
acct.addFunds(34);

// Beachten Sie, dass subFunds in der externen JAR-Datei definiert ist als\:
// public int subFunds (int amnt)
acct.subFunds(10);

// Definieren Sie Ihre eigene Methode
String getStatement(MyAccount acc) {
    StringBuffer strBuff = new StringBuffer("Account Balance: " + acc.getBalance());
    Date dt = new Date(System.currentTimeMillis());
    strBuff.append(" on date: " + dt);
    return strBuff.toString();
}

// Verwenden Sie die von Ihnen definierte Methode und drucken Sie die Aussage unter Verwendung des
Objekts 'logger' aus,
// das Sie auf der Seite 'Logger' des Operators festlegen.
logger.info(getStatement(acct));

// Am Ende der Ausführung des Operators\:
// acct wird im Datensatz des Operators als Variable des Typs 'JavaObject' gespeichert.
// Die Logger-Protokolldatei enthält folgende Meldung\:
//      Account Balance: 124 on date: Thu Aug 22 11:27:29 EDT 2013
//      (Die Meldung gibt das korrekte Datum und die Uhrzeit der Ausführung wieder).
```

Ressource für die Ausführung des Operators "Java aufrufen" - Beispiel

Der Installationsprozess fügt dem Ordner "Anwenderressource" eine Ressource unter "Repository" im Auswahlmenü "Anwenderressourcen verwalten" auf der Registerkarte "Konfiguration" hinzu. Die JAR-Datei "MyAccount.jar" befindet sich im Ordner "Invoke_Java_Op_Example_jars". Sie können die Datei "MyAccount.jar" verwenden, um das Java-Beispiel auszuführen, das im Feld "Erforderliche Hauptmethode" des Operators "Java aufrufen" angegeben ist.



Operator-Ports

Erfolg

Der Operator wird erfolgreich abgeschlossen.

Fehlschlag

Der Operator schlägt aus folgenden Gründen fehl:

- Sie verwenden einen BeanShell-Befehl im Operator-Code, wenn festgelegt ist, dass der Operator im "Strict Java Mode" ausgeführt wird. Ein BeanShell-Befehl wird möglicherweise nicht im "Strict Java Mode" unterstützt.
- Sie verwenden nicht eingegebene Variablendeklarationen im Operator-Code, wenn festgelegt ist, dass der Operator im "Strict Java Mode" ausgeführt wird.
- Ein Fehler tritt aufgrund des ausgeführten Codes auf. Zum Beispiel wird auf einem Java Objekt die falsche Methode aufgerufen.
- Der Operator-Code gibt eine Ausnahme aus, während er ausgeführt wurde.
- Sie versuchen die Protokollierung im Code des Operators zu verwenden, während der Operator nicht zum Verwenden der Protokollierung eingestellt ist. Die Protokollierung ist nicht im Zusammenhang mit dem Operator-Code definiert, und der Operator kann keine Methoden der Protokollierung lösen.
- Die Protokollierung wird zur Verwendung einer schreibgeschützten Datei konfiguriert.
- Die Protokollierung wird zur Verwendung einer Protokolldatei konfiguriert, die eigentlich ein Verzeichnis ist.

- Sie geben einen ungültigen Pfad in die externen JAR-Dateien der Operatorkategorie ein.
- Das Java-Objekt, das CA Process Automation versucht im Operator am Ende der Ausführung zu speichern, kann nicht serialisiert werden. Der Operator schlägt mit einer Fehlermeldung fehl.
- Sie übergeben eine Datensatzvariable des Typs "JavaObject" an den Operator. Allerdings geben Sie nicht die JAR-Datei an, in der sich die Klassendefinition des Java-Objekts befindet. Ein Fehler zeigt an, dass der Operator beim Deserialisieren des Objekts den Deskriptor der Klasse nicht lesen konnte.
- Sie übergeben eine leere Datensatzvariable des Typs: JavaObject (Wert: "[JavaObject]") an den Operator. Der Operator schlägt fehl und zeigt an, dass die Deserialisierung des Objekts "Null" fehlgeschlagen ist.
- Die Liste der JAR-Dateien für den Operator enthalten eine nicht vorhandene JAR-Datei.
- Die Liste der JAR-Dateien für die Operatorkategorie enthält eine nicht vorhandene JAR-Datei.
- Der Operator "Java aufrufen" kann nicht auf einem Koordinationsrechner ausgeführt werden. Wenn die Eingabe im Feld "Ziel" zu einem Koordinationsrechner aufgelöst wird, dann schlägt der Operator mit folgender Ausgabevariablen fehl:
  - **ResponseMessage**=Der Operator "Java aufrufen" kann nicht auf einem Koordinationsrechner ausgeführt werden.
  - **ResponseCode**=SYSTEM_ERROR
- Sie haben keinen Java-Code in der Hauptmethode des Operators angegeben. Die Hauptmethode besteht aus normalen Java-Anweisungen und Ausdrücken. Sie müssen Objekte initialisieren und verwenden, die im Java-SDK oder in externen JAR-Dateien definiert sind.

  Der Operator "Java aufrufen" wird normalerweise so verwendet, dass Sie externe JAR-Dateipfade im Operator oder im Modul angeben. Sie müssen auch einen Java-Code in der Hauptmethode des Operators angeben. Dieser Code verwendet die Objekte, die Sie in den externen JAR-Dateien definiert haben.

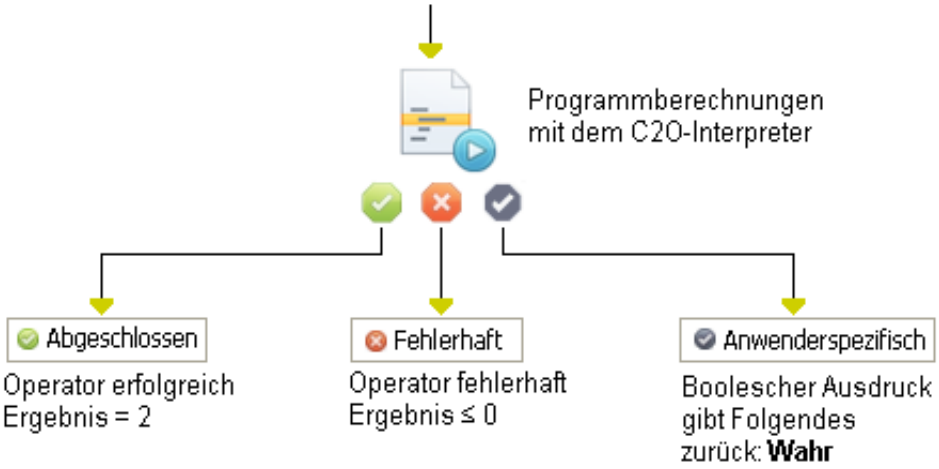
Anwenderspezifische Ports

Ist verfügbar, wenn während des Prozessdesigns durch den Anwender festgelegt.

Operator "JavaScript ausführen"

Der Operator "JavaScript ausführen" führt Berechnungen und Zuweisungen der Datensatzvariablen aus. Der Operator führt die folgenden Aktionen aus:

- Der Operator interpretiert JavaScript-Anweisungen in seinem Quellcode.
- Ermöglicht Berechnungen, um Werte für Variablen festzulegen. Diese Werte können dann für Parametereinstellungen in nachfolgenden Operatoren im gleichen Prozess oder in anderen Prozessen verwendet werden.



Eingabeparameter

SourceCode

Öffnet den Code-Editor.

Verwenden Sie den Code-Editor, um eine oder mehrere JavaScript-Anweisungen festzulegen. Jede Anweisung endet mit einem Semikolon (;). Die folgenden Anweisungen legen beispielsweise die Variablen für Tag, Monat und Jahr in einem benannten Datensatz fest:

```
Datasets["/exploit/variables/date"].day = "31";
Datasets["/exploit/variables/date"].month = "Juli";
Datasets["/exploit/variables/date"].year = "2013";
```

Hinweis: Weitere Informationen über die Verwendung des CA Process Automation Code-Editors finden Sie im *Handbuch für Inhaltsdesign*.

Ausgabeparameter

SourceCode

Kapitel 15: Webservices

Die Webservices-Operatoren unterstützen Anrufe zu Remote-Services mittels SOAP oder XML. Diese Operatoren rufen außerdem Antworten ab und speichern die Informationen der Antwort, damit sie von anderen Operatoren in einem Prozess verwendet werden können.

Die Webservices-Operatoren stellen auch Datenverwaltungseinrichtungen über ein Netzwerk bereit, das standardmäßige und weit verbreitete Protokolle wie HTTP verwendet. Unterstützung für RESTful-Services wird auch über die HTTP-Operatoren bereitgestellt.

HTTP-Operatoren: Allgemeine Eingabeparameter

Die Eingabeparameter, die für alle HTTP-Operatoren gelten, werden folgendermaßen in separate Kategorien eingeteilt:

- Informationen zu HTTP-URL (siehe Seite 440)
- HTTP-Proxy-Informationen (siehe Seite 444)
- Informationen zu HTTP-Header (siehe Seite 448)
- Informationen zu HTTP-Cookies (siehe Seite 449)
- Informationen zum Inhalt der HTTP-Antwort (siehe Seite 449)
- Informationen zur HTTP-Konfiguration (siehe Seite 451)

Die Eigenschaften der HTTP-Operatoren benötigen folgende Eingabeparameter:

- HTTP DELETE - Nur Allgemein
- HTTP GET - Allgemein und Informationen zu GET (siehe Seite 464)
- HTTP HEAD - Allgemein und Informationen zu HEAD (siehe Seite 468)
- HTTP OPTIONS - Nur Allgemein
- HTTP POST - Allgemein und Informationen zu POST (siehe Seite 478)
- HTTP POST FORM - Allgemein und Informationen zu POST FORM (siehe Seite 484)
- HTTP PUT - Allgemein und Informationen zu PUT (siehe Seite 492)
- HTTP TRACE - Nur Allgemein

Hinweis: Wenn nicht anders angegeben, überschreiben Feldeingaben die entsprechenden Feldwerte, die von der Konfiguration der Operatorkategorieebene geerbt wurden.

Informationen zu HTTP-URL

Zu den Informationen zu HTTP-URL gehören Eingabeparameter, die für folgende Operatoren gelten:

- HTTP Delete
- HTTP Get
- HTTP Head
- HTTP Options
- HTTP Post
- HTTP Post to Form
- HTTP Put
- HTTP Trace

URL

Gibt die URL der HTTP-Anforderung an. Die URL beginnt mit http:// oder https://.

SSL-Zertifikat validieren?

Gibt an, ob ein gültiges SSL-Zertifikat gefunden wird. Dieses Feld ist relevant, wenn eine HTTPS-URL angefordert wird.

Werte

- Wahr: Validiert das SSL-Zertifikat und lässt den Operator fehlschlagen, wenn das Zertifikat ungültig ist.
- Falsch: Akzeptiert das SSL-Zertifikat, auch wenn es ungültig ist, und fährt mit dem HTTP-Aufruf fort.
- Leer: Verwendet den Wert, der auf der Ebene der Operatorkategorie für "SSL-Zertifikat standardmäßig validieren?" festgelegt wurde.

Authentifizierungstyp

Aktivieren Sie eine der folgenden Optionen.

HTTP-Authentifizierung

Gibt an, ob der HTTP-Server in der angegebenen URL Authentifizierung benötigt. Die HTTP-Operatoren unterstützen HTTP-Basisauthentifizierung, präemptive HTTP-Authentifizierung und NTLM-Authentifizierung. Wenn beide Authentifizierungstypen erforderlich sind, legen Sie diesen Wert auf "Wahr" fest.

Werte

- Wahr: Gibt an, dass der HTTP-Server Authentifizierung benötigt.
- Falsch: Gibt an, dass der HTTP-Server keine Authentifizierung benötigt.
- Leer: Gibt an, dass der Wert verwendet werden soll, der auf der Ebene der Operatorkategorie festgelegt ist.
- Anderer Wert: Entspricht dem Wert "Falsch".

Standard

Leer: Gibt an, dass der Wert verwendet werden soll, der auf der Ebene der Operatorkategorie festgelegt ist.

Hinweise

- Wenn HTTP-Authentifizierung im Operator auf "Falsch" festgelegt ist, dann werden "NTLM-Authentifizierung", "Anwendername", "Kennwort" und "Domänenname" deaktiviert.
- Wenn HTTP-Authentifizierung im Operator nicht auf "Falsch" festgelegt ist ("Wahr", "Datensatzvariable" oder ein anderer Wert), dann wird Folgendes aktiviert:

  - NTLM-Authentifizierung

  - Anwendername

  - Kennwort

  - Domänenname

Präemptive HTTP-Authentifizierung

Bei der Interaktion mit einem Webservice agieren HTTP-Operatoren von CA Process Automation üblicherweise wie Webbrowser. Als Erstes wird dabei mit dem Webservice vereinbart, welches Authentifizierungsschema verwendet werden soll. Diese Vereinbarung wird getroffen, bevor der Operator seine Anfrage an den Webservice stellt.

Bei HTTP-Basisauthentifizierung gibt der Webservice üblicherweise im Zuge der Authentifizierungsvereinbarung mit dem CA Process Automation-Operator an, dass Basisauthentifizierung erforderlich ist. Bei einer Konfiguration für HTTP-Authentifizierung vereinbart der CA Process Automation-Operator mit dem Webservice, dass Basisauthentifizierung verwendet wird.

Manche Webservices geben jedoch im Zuge der Vereinbarung nicht an, dass sie Basisauthentifizierung benötigen. Stattdessen nehmen sie an, dass der Operator von sich aus Basisauthentifizierung sendet. Diese Vorgangsweise wird normalerweise verwendet, um zusätzlichen Verbindungsaufwand zu reduzieren. In solchen Fällen sollte der HTTP-Operator von CA Process Automation so konfiguriert werden, dass präemptive HTTP-Authentifizierung (anstelle von HTTP-Authentifizierung) verwendet wird. Dadurch wird der Operator aufgefordert, HTTP-Basisauthentifizierung ohne Negotiation an den Webservice zu senden.

NTLM-Authentifizierung

Gibt an, ob der HTTP-Server in der angegebenen URL NTLM-Authentifizierung benötigt. CA Process Automation verwendet HTTP-Basisauthentifizierung, wenn NTLM-Authentifizierung nicht ausgewählt ist.

Werte

- Wahr: Gibt an, dass der HTTP-Server NTLM-Authentifizierung benötigt.
- Falsch: Gibt an, dass der HTTP-Server keine NTLM-Authentifizierung benötigt. Der Server verwendet HTTP-Basisauthentifizierung
- Leer: Gibt an, dass der Wert verwendet werden soll, der auf der Ebene der Operatorkategorie festgelegt ist.
- Anderer Wert: Entspricht dem Wert "Falsch".

Anwendername:

Gibt den Anwendernamen an, der für die Authentifizierung gegen die angegebene URL verwendet wird.

Kennwort

Gibt das Kennwort für den angegebenen Anwendernamen an.

Domänenname

Gibt den Namen der Domäne an, der für die Authentifizierung gegen die angegebene URL verwendet wird.

Verwenden Sie folgende Richtlinien:

- Geben Sie den Domänennamen (erforderlich) ein, wenn der Operator NTLM-Authentifizierung verwendet.
- Lassen Sie dieses Feld leer, wenn der Domänenname für die Authentifizierung nicht erforderlich ist. Ein Domänenname ist möglicherweise nicht erforderlich, wenn der Operator HTTP-Basisauthentifizierung verwendet.

Verwendungshinweise für "Domänenname", "Anwendername" und "NTLM-Authentifizierung"

Ein leeres Feld "Domänenname" fordert nicht automatisch den Operator auf, den Domänennamenwert aus der Operatorkategorie zu erben. Das Feld "Domänenname" ist folgendermaßen mit dem Feld "Anwendername" verbunden:

- Wenn der Domänenname für den Operator angegeben ist, verwendet der Operator ihn.
- Wenn der Domänenname des Operators leer ist und der Anwendername des Operators angegeben ist (nicht leer), verwendet der Operator einen leeren Domänennamen.

Der Operator verwendet den standardmäßigen Domänennamen aus der Operatorkategorie, wenn folgende Felder leer sind (nicht angegeben):

- Der Domänenname des Operators.
- Der Anwendername für den Operator.

Ein angegebener Domänenname wird folgendermaßen verwendet:

- Wenn der Operator NTLM-Authentifizierung verwendet, wird der Domänenname wie angegeben verwendet, ohne dass er an den Anwendernamen angehängt wird.
- Wenn der Operator HTTP-Basisauthentifizierung verwendet, wird der Domänenname an den Anwendernamen folgendermaßen angehängt: Anwendernamen = Anwendername@Domänenname

HTTP-Proxy-Informationen

Zu den HTTP-Proxy-Informationen gehören Eingabeparameter, die für folgende Operatoren gelten:

- HTTP Delete
- HTTP Get
- HTTP Head
- HTTP Options
- HTTP Post
- HTTP Post to Form
- HTTP Put
- HTTP Trace

Hinweis: Wenn nicht anders angegeben, überschreiben Feldeingaben die entsprechenden Feldwerte, die von der Konfiguration der Operatorkategorieebene geerbt wurden.

Proxy verwenden?

Gibt an, ob die HTTP-Anrufe durch einen Proxy-Server gehen. Dieses Feld überschreibt das Modulfeld. Wenn dieses Feld leer gelassen wird, verwendet der Operator den Standardwert, der auf der Modulebene festgelegt ist.

Werte

Eine der folgenden Komponenten:

- Wahr: Zeigt an, dass HTTP-Aufrufe durch einen Proxy-Server weitergeleitet werden sollen.
- Falsch: Zeigt an, dass HTTP-Aufrufe nicht durch einen Proxy-Server weitergeleitet werden sollen.
- Leer: Zeigt an, dass der Wert verwendet werden soll, der auf der Modulebene festgelegt ist.
- Anderer Wert: Entspricht dem Wert "Falsch".

Hinweise:

- Wenn dieses Feld auf "Falsch" festgelegt wird, werden die verbleibenden Felder in den HTTP-Proxy-Informationen deaktiviert.
- Wenn dieses Feld im Operator nicht auf "Falsch" festgelegt wird ("Wahr", "Datensatzvariable" oder ein anderer Wert), werden die verbleibenden Felder in den HTTP-Proxy-Informationen aktiviert.

Proxy-Host

Gibt entweder die URL (mit http oder https) des Proxy-Servers oder den FQDN des Proxy-Servers an.

Hinweis: Wenn der FQDN eingegeben ist, wird das HTTP-Schema verwendet, um den Proxy-Server unter http://<FQDN von proxy>:<port> zu kontaktieren.

Proxy-Port

Gibt den Port des angegebenen Proxy-Hosts an.

Werte

Eine der folgenden Komponenten:

- Leer: Erbt den Wert "Standardmäßiger Proxy-Port", der auf der Modulebene festgelegt ist, sofern vorhanden. Anderenfalls wird Port 80 verwendet.
- Die angegebene Portnummer.

Proxy-Authentifizierung?

Gibt an, ob der Proxy-Server in der angegebenen Proxy-URL Authentifizierung benötigt. Proxy-Authentifizierung kann entweder HTTP-Basisauthentifizierung oder NTLM-Authentifizierung sein. Wenn beide Authentifizierungstypen erforderlich sind, legen Sie diesen Wert auf "Wahr" fest.

Werte

Eine der folgenden Komponenten:

- Wahr: Zeigt an, dass der Proxy-Server Authentifizierung benötigt.
- Falsch: Zeigt an, dass der Proxy-Server keine Authentifizierung benötigt.
- Leer: Zeigt an, dass der standardmäßige Wert verwendet werden soll, der auf der Modulebene festgelegt ist.
- Anderer Wert: Entspricht dem Wert "Falsch".

Proxy-NTLM-Authentifizierung?

Zeigt an, ob der angegebene Proxy-Host NTLM-Authentifizierung benötigt.

Werte

Eine der folgenden Komponenten:

- Wahr: Zeigt an, dass der angegebene Proxy-Host NTLM-Authentifizierung benötigt.
- Falsch: Zeigt an, dass der angegebene Proxy-Host keine NTLM-Authentifizierung benötigt. Der Proxy-Host verwendet HTTP-Basisauthentifizierung.
- Leer: Gibt an, dass der Wert verwendet werden soll, der auf der Modulebene festgelegt ist.
- Anderer Wert: Entspricht dem Wert "Falsch".

Hinweis

Wenn Proxy-Authentifizierung im Operator auf "Falsch" festgelegt ist, dann wird Folgendes deaktiviert:

- Proxy-NTLM-Authentifizierung
- Proxy-Anwendername
- Proxy-Kennwort
- Name der Proxy-Domäne

Wenn Proxy-Authentifizierung im Operator nicht auf "Falsch" festgelegt ist ("Wahr", "Datensatzvariable" oder ein anderer Wert), dann wird Folgendes aktiviert:

- Proxy-NTLM-Authentifizierung
- Proxy-Anwendername
- Proxy-Kennwort
- Name der Proxy-Domäne

Proxy-Anwendername

Gibt den Anwendernamen an, der für die Authentifizierung mit dem Proxy verwendet wird.

Proxy-Kennwort

Gibt das Kennwort an, das mit dem Proxy-Anwendernamen verbunden ist.

Name der Proxy-Domäne

Gibt den Namen der Domäne an, der für die Authentifizierung gegen den angegebenen Proxy-Server verwendet wird.

Verwenden Sie folgende Richtlinien:

- Geben Sie den Namen der Proxy-Domäne (erforderlich) ein, wenn der Operator NTLM-Authentifizierung gegen den Proxy-Server verwendet.
- Lassen Sie dieses Feld leer, wenn der Name der Proxy-Domäne für die Authentifizierung nicht erforderlich ist. Wenn der Operator HTTP-Basisauthentifizierung gegen den Proxy-Server verwendet, ist normalerweise kein Domänenname erforderlich.

Verwendungshinweise für "Name der Proxy-Domäne", "Proxy-Anwendername" und "Proxy-NTLM-Authentifizierung"

- Ein leeres Feld in "Name der Proxy-Domäne" fordert nicht automatisch den Operator auf, den Wert "Name der Proxy-Domäne" aus dem Modul zu erben.
- Das Feld "Name der Proxy-Domäne" des Operators wird folgendermaßen mit dem Feld "Proxy-Anwendername" des Operators verbunden:
  - Wenn der Name der Proxy-Domäne des Operators angegeben ist, verwendet der Operator diesen Namen.
  - Wenn "Name der Proxy-Domäne" leer ist und "Proxy-Anwendername" angegeben ist (nicht leer), verwendet der Operator einen leeren Namen der Proxy-Domäne.
  - Wenn "Name der Proxy-Domäne" leer ist und "Proxy-Anwendername" nicht angegeben ist (leer), verwendet der Operator den geerbten Wert in "Standardmäßige Proxy-Domänenname".
- Ein angegebener Name der Proxy-Domäne wird folgendermaßen verwendet:
  - Wenn der Operator NTLM-Authentifizierung gegen den Proxy-Server verwendet, wird der angegebene Name der Proxy-Domäne wie angegeben verwendet. Der Name der Proxy-Domäne wird nicht an den Proxy-Anwendernamen angehängt.
  - Wenn der Operator HTTP-Basisauthentifizierung gegen den Proxy-Server verwendet, wird der angegebene Name der Proxy-Domäne folgendermaßen an den Proxy-Anwendernamen angehängt:

    ```
    Anwendername = Anwendername@Domänenname
    ```

Informationen zu HTTP-Header

Indizierte ValueMaps für HTTP-Header verwenden?

Gibt an, ob ein indiziertes ValueMap für die Header von HTTP-Anforderungen verwendet werden soll.

Werte:

- Ausgewählt: Zeigt an, dass Header von HTTP-Anforderungen als indiziertes ValueMap im Feld "HTTP-Header - Indiziertes ValueMap" angegeben werden soll.
- Deaktiviert: Zeigt an, dass Header von HTTP-Anforderungen im Feld "HTTP-Header" eingegeben werden soll.

HTTP-Header

Gibt die Namen der HTTP-Header in der Spalte "Schlüssel" und die Werte der HTTP-Header in der Spalte "Wert" an. Headers müssen im US-ASCII-Format sein.

Verwenden Sie die Schaltflächen, um Header hinzuzufügen, zu entfernen oder neu anzuordnen.

Hinweis: Der Operator ignoriert Headers, bei denen der Schlüssel leer ist, das heißt, bei denen kein Header-Name angegeben ist.

HTTP-Header - Indiziertes ValueMap

Gibt den Namen eines indizierten ValueMap an, das die Namen und die entsprechenden Werte der HTTP-Header enthält. Das indizierte ValueMap muss das gleiche Format haben wie das ValueMap, das im Feld "HTTP-Header" aufgelistet ist. Das indizierte ValueMap muss sowohl den Parameter "Schlüssel" als auch den Parameter "Wert" haben.

Hinweis: Der Operator ignoriert Headers, bei denen der Schlüssel leer ist, das heißt, bei denen kein Header-Name angegeben ist.

Informationen zu HTTP-Cookies

HTTP-Cookies-Speicher - Indiziertes ValueMap

Geben Sie ein indiziertes ValueMap ein, das die HTTP-Cookies enthält, die in diesem Operator festgelegt werden sollen.

Dieses Feld aktiviert die HTTP-Statusverwaltung, indem Anwender "HTTPCookiesStore" von einem Operator an einen anderen übergeben können, der auf die gleiche Cookiedomäne gerichtet ist. Das indizierte ValueMap muss das gleiche Format wie das ValueMap haben, das von anderen HTTP-Operatoren in der Ausgabevariablen "HTTPCookiesStore" zurückgegeben wurde. Normalerweise hat dieses Feld folgendes Format:

```
PreviousHttpOperator.HTTPCookiesStore
```

Durch das Abrufen von "HTTPCookiesStore" eines anderen Operators, kann dieser Operator anwendbare Cookies senden. Zu den anwendbaren Cookies gehören jene, die in der HTTP-Anforderung oder in der HTTP-Antwort des früheren Operators festgelegt wurden. Dieser Operator sendet nur die nicht abgelaufenen Cookies (vom HTTPCookiesStore), deren Attribute auf die URL dieses Operators angewendet werden können. Beispiel-Attribute schließen Domäne, Pfad und isSecure ein.

Informationen zum Inhalt der HTTP-Antwort

Inhalt der HTTP-Antwort in Datei speichern?

Gibt an, ob der Text der HTTP-Antwortmeldung in eine Datei gespeichert werden soll. Wählen Sie dieses Feld aus, um das Feld "Pfad der Inhaltsdatei der HTTP-Antwort" und "Bei Textantwort mit Kodierung speichern" zu aktivieren.

Werte

- Ausgewählt: Speichert den Text der HTTP-Antwortmeldung in eine Datei.
- Deaktiviert: Der Text der HTTP-Antwortmeldung wird nicht in einer Datei gespeichert.

Pfad der Inhaltsdatei der HTTP-Antwort

Gibt an, wo der Text der HTTP-Antwortmeldung gespeichert werden soll. Geben Sie den Pfad der lokalen Datei auf dem Host ein, auf dem der Kontaktpunkt ausgeführt wird.

Wenn Antwortdatei vorhanden ist?

Gibt die Aktion an, die durchgeführt werden soll, wenn die Antwortdatei vorhanden ist. Verfügbare Aktionen sind das Erstellen einer Datei oder das Überschreiben des Inhalts der vorhandenen Datei.

Der Pfad der Datei wird in der Ausgabevariable des Operators "HTTPResponseContentFilePath" aufgelistet.

Werte

Dieser Wert kann Folgendes sein:

- createFile: Zeigt an, dass eine Datei erstellt werden soll.
- overwriteFile: Zeigt an, dass der Inhalt der Antwortdatei mit dem neuen HTTP-Antworttext überschrieben werden soll.
- Leer: Entspricht dem Wert "createFile".
- Eine andere Ganzzahl: Entspricht dem Wert "createFile".

Bei Textantwort mit Kodierung speichern

Geben Sie diese Codierung an, wenn Sie eine Textantwort erwarten. Der Inhaltstyp der Antwort hat folgendes Format:

`text/XXXX`

Diese Codierung wird verwendet, um die Antwort in der Antwortdatei zu schreiben.

Wenn die erhaltene Antwort nicht das Format "text/XXXX" hat, dann wird dieses Feld ignoriert.

Werte

Dieser Wert kann Folgendes sein:

- 0 : Festgelegt im Header der HTTP-Antwort - Geben Sie 0 ein, um die Kodierung zu verwenden, die im Header der HTTP-Antwort festgelegt ist.
- 1 : Standardmäßige Kodierung von PAM - Geben Sie 1 ein, um die Standard-Systemkodierung von CA Process Automation zu verwenden.
- 2 : Kodierung in 'Vom Anwender festgelegte Textantwortkodierung' festlegen - Geben Sie 2 ein, um eine (zu verwendende) Kodierung im Feld "Vom Anwender festgelegte Textantwortkodierung" festzulegen.
- Leer: Fordert den Operator auf, den Wert 0 zu verwenden (Kodierung verwenden, die im Header der HTTP-Antwort festgelegt ist).
- Eine andere Ganzzahl: Fordert den Operator auf, den Wert 0 zu verwenden (Kodierung verwenden, die im Header der HTTP-Antwort festgelegt ist).

Wenn dieses Feld auf 0 oder 1 festgelegt ist, dann ist das Feld "Vom Anwender festgelegte Textantwortkodierung" deaktiviert.

Vom Anwender festgelegte Textantwortkodierung

Geben Sie eine Kodierung an, die verwendet werden soll, wenn die Textantwort in die Antwortdatei geschrieben wird.

Inhalt der HTTP-Antwort in Datensatzvariable speichern?

Gibt an, ob der Text der HTTP-Antwortmeldung in die Variable "HHTPResponseContent" im Datensatz des Operators gespeichert werden soll. Wenn Speichern aktiviert ist, wird das Feld "Größenbeschränkung für die Datensatzvariable der HTTP-Antwort" aktiviert.

Werte

- Ausgewählt: Speichert den Text des Inhalts der HTTP-Antwort in die Variable "HHTPResponseContent" im Datensatz des Operators.
- Deaktiviert: Der Inhalt der HTTP-Antwort wird nicht gespeichert.

Größenbeschränkung für die Datensatzvariable der HTTP-Antwort (Byte)

Gibt die Höchstanzahl von Byte (vom Text der HTTP-Antwortmeldung) an, die in der Datensatzvariable "HTTPResponseContent" des Operators gespeichert wird.

Wert

Geben Sie einen numerischen Wert ein.

Standard

4096 Byte (wenn dieses Feld leer gelassen wird)

Informationen zur HTTP-Konfiguration

HTTP Version (Produktversion)

Gibt die Version des HTTP-Protokolls an.

Werte:

Eine der folgenden Komponenten:

- 1.0 - Zeigt an, dass der Operator die Version 1.0 des HTTP-Protokolls verwenden soll.
- 1.1 - Zeigt an, dass der Operator die Version 1.1 des HTTP-Protokolls verwenden soll.
- Leer: Zeigt an, dass der Operator den Wert verwenden soll, der auf der Ebene der Operatorkategorie festgelegt ist. Ein leerer Wert oder ein anderer Wert als 1.1 oder 1.0 auf der Kategorieebene, fordert den Operator auf, 1.1 zu verwenden.
- Ein anderer Wert als 1.0 oder 1.1 - HTTP-Protokoll 1.1 wird verwendet

Standard:

Leer

Zeitlimit für Verbindung (Sek.)

Gibt die Zeit an, die maximal gewartet wird, um eine HTTP-Verbindung zu erstellen, bevor der Operator das Zeitlimit erreicht.

Werte:

Eine der folgenden Komponenten:

- Ein numerischer Wert, der das Zeitlimit der Verbindung in Sekunden anzeigt.
- "0" zeigt kein Zeitlimit an, d. h. null Sekunden.
- Leer: Zeigt das "Standardmäßige Zeitlimit für Verbindung" an, das auf der Ebene der Operatorkategorie (sofern vorhanden) festgelegt ist. Andernfalls werden 0 Sekunden angezeigt.

Standard:

Leer.

Zeitlimit für Socket (Sek.)

Gibt die Zeit an, die zwischen zwei aufeinanderfolgenden Paketen von HTTP-Antwortdaten maximal gewartet wird.

Werte:

Eine der folgenden Komponenten:

- Ein numerischer Wert, der das Zeitlimit für Socket in Sekunden anzeigt.
- "0" zeigt kein Zeitlimit an, d. h. null Sekunden.
- Leer: Zeigt das "Standardmäßige Socket-Zeitlimit" an, das auf der Ebene der Operatorkategorie (sofern vorhanden) festgelegt ist. Andernfalls werden 0 Sekunden angezeigt.

Standard:

Leer.

Umleitungen bearbeiten?

Gibt an, ob Umleitungen automatisch bearbeitet werden sollen.

Werte:

Eine der folgenden Komponenten:

- Wahr: Umleitungen werden automatisch bearbeitet.
- Falsch: Umleitungen werden nicht automatisch bearbeitet.
- Leer: Der Wert "Umleitungen standardmäßig bearbeiten?" wird verwendet, Wert, der auf der Ebene der Operatorkategorie festgelegt ist.
- Anderer Wert: Gibt den Wert "Falsch" an.

Standard:

Leer.

Höchstanzahl von Umleitungen

Gibt die Höchstanzahl an Umleitungen an, die beachtet werden sollen, wenn "Umleitungen bearbeiten?" auf "Wahr" festgelegt ist.

Werte:

Eine der folgenden Komponenten:

- Ein numerischer Wert gibt die Höchstanzahl der erlaubten Umleitungen an.
- Leer: Standardmäßige Höchstanzahl von Umleitungen, wenn festgelegt. Anderenfalls wird "100" verwendet.

Standard:

Leer.

HTTP-Operatoren: Allgemeine Ausgabeparameter

Ausgabevariablen enthalten keine Daten, wenn der Operator keine HTTP-Antwort enthält, weil einer der folgenden Fehler auftritt:

- Eingabe enthält eine unbekannte URL.
- Die HTTP-Verbindung erreicht das Zeitlimit.
- Das Socket erreicht ein Zeitlimit.

HTTPRequestUrl

Gibt die URL, einschließlich aller URL-Parameter, der HTTP-Anforderung an.

HTTPResponseStatusLine

Gibt die Statuszeile der HTTP-Antwort an. Die Statuszeile ist die erste Zeile der HTTP-Antwortmeldung. Die Statuszeile besteht aus der Protokollversion, dem Statuscode und der zugeordneten Grundphrase.

HTTPResponseStatusCode

Gibt den Statuscode der HTTP-Antwort an. Je nach Statuscode schlägt der Operator fehl oder er ist erfolgreich.

- Der Operator schlägt fehl, wenn der Statuscode größer oder gleich 300 ist.
- Der Operator ist erfolgreich, wenn der Statuscode kleiner als 300 ist.

HTTPResponseReasonPhrase

Gibt die Grundphrase der HTTP-Antwort an.

HTTPResponseProtocolVersion

Gibt die Protokoll-Version der HTTP-Antwort an.

HTTPResponseContentType

Gibt den Inhaltstyp-Header des Inhalts der HTTP-Antwort an.

HTTPResponseContentCharset

Gibt die Zeichenkodierung des Inhalts der HTTP-Antwort an. Diese Zeichenkodierung ist Teil des Inhaltstyp-Header und wird in folgender Form angezeigt:

`“content-type= xxxxx; charset=xxxx”`

Dieser Zeichensatz wird nur festgelegt, wenn der Inhaltstyp nur Zeichen wie z. B. Text/xxx umfasst.

HTTPResponseContentLength

Gibt die Anzahl der Byte des Inhalts der HTTP-Antwort an. Eine negative Zahl bedeutet, dass die Inhaltslänge nicht bekannt ist.

HTTPResponseContentEncoding

Gibt die Inhaltskodierung vom Header des Inhalts der HTTP-Antwort an. Ein leerer Wert zeigt an, dass die Inhaltskodierung unbekannt ist.

HTTPResponseContentIsChunked

"Wahr" zeigt an, dass der Inhalt der HTTP-Antwort mit aufgeteilter Kodierung empfangen wurde. "Falsch" wird zurückgegeben, wenn die Bedingung für "Wahr" nicht erfüllt ist.

HTTPResponseContentFilePath

Gibt den Pfad zur Datei an, in der der Inhalt der HTTP-Antwort gespeichert wurde. Ein leerer Wert zeigt an, dass der Operator nicht zum Speichern des Inhalts der HTTP-Antwort (Meldungstext) in einer Datei eingerichtet ist.

- Wenn die Eingabe für "Wenn Antwortdatei vorhanden ist?" 0 war und der Dateipfad als Eingabe in "Pfad der Inhaltsdatei der HTTP-Antwort" vorhanden ist, dann enthält das Feld "HTTPResponseContentFilePath" den Pfad zu der neuen Datei, in der der Inhalt der HTTP-Antwort gespeichert wurde.
- Wenn die Eingabe für "Wenn Antwortdatei vorhanden ist?" 1 war und der Dateipfad als Eingabe in "Pfad der Inhaltsdatei der HTTP-Antwort" vorhanden ist, dann enthält das Feld "HTTPResponseContentFilePath" den Pfad, der in "Pfad der Inhaltsdatei der HTTP-Antwort" angegeben ist.

HTTPResponseContent

Gibt den Inhalt der HTTP-Antwort bis zur Anzahl der Byte an, die im Feld "Größenbeschränkung für die Datensatzvariable der HTTP-Antwort (Byte)" angegeben sind. Ein leerer Wert kann anzeigen, dass der Operator nicht zum Speichern des Inhalts der HTTP-Antwort (Meldungstext) in seinem Datensatz eingerichtet ist. Ein leerer Wert kann auch anzeigen, dass der Inhalt der HTTP-Antwort leer ist.

HTTPResponseHtmlContent

Gibt den Inhalt der HTTP-Antwort an, der als HTML im Datensatz des Operators wiedergegeben wurde. Die Inhaltstyp-Header, der mit "Text/html" beginnt, zeigt an, dass der Inhalt der HTTP-Antwort HTML ist. Wenn CA Process Automation erkennt, dass der Inhalt der HTTP-Antwort HTML ist, wird der Inhalt der HTTP-Antwort im Datensatz des Operators als HTML wiedergegeben. Die Rohdaten sind für JavaScript-Code in "HTTPResponseContent" weiterhin zugreifbar. Ein leerer Wert kann anzeigen, dass der Operator nicht zum Speichern des Inhalts der HTTP-Antwort (Meldungstext) in einem Datensatz eingerichtet ist. Ein leerer Wert kann auch bedeuten, dass CA Process Automation erkennt, dass der Inhalt der HTTP-Antwort nicht HTML oder leer ist.

Hinweis: CA Process Automation gibt nur grundlegende HTML-Seiten wieder. CA Process Automation gibt keine komplexen HTML-Seiten wieder.

HTTPResponseHeaders

Gibt die HTTP-Header der HTTP-Antwort an. Die Header werden als indiziertes ValueMap zurückgegeben, wobei jedes ValueMap einen einzelnen Header und die folgenden zwei Parameter enthält:

Schlüssel

Gibt den Namen des HTTP-Header an.

Wert

Gibt den Wert des HTTP-Header an.

HTTPRequestHeaders

Gibt die HTTP-Header der gesendeten HTTP-Anforderung an. Dieses Feld enthält die HTTP-Header, die als Eingabe in den Feldern "HTTP-Header" oder "HTTP-Header - Indiziertes ValueMap" des Operators eingegeben sind. Dieses Feld enthält auch die HTTP-Header für Authentifizierung, Proxy und andere, die der Operator hinzugefügt hat, bevor er die Anfrage sendet.

Die Header werden als indiziertes ValueMap zurückgegeben, wobei jedes ValueMap einen einzelnen Header und die folgenden Parameter enthält:

Schlüssel

Gibt den Namen des HTTP-Header an.

Wert

Gibt den Wert des HTTP-Header an.

HTTPRequestLine

Gibt die Anforderungszeile der gesendeten HTTP-Anforderung an. Die Zeile der HTTP-Anforderung enthält die HTTP-Methode, die URL und die HTTP-Version.

HTTPCookiesStore

Gibt die analysierte Version der HTTP-Cookies an, die in der Anforderung gesendet wurde, und gibt die HTTP-Cookies an, die in den Antwort-Header eingebettet sind. Die Cookies werden als indiziertes ValueMap zurückgegeben, wobei jedes ValueMap einen einzelnen Cookie mit den folgenden definierten Parametern enthält:

Name

Gibt den Namen dieses HTTP-Cookies an.

Wert

Gibt den Wert dieses HTTP-Cookies an.

Version

Gibt die Version der Cookie-Spezifikation an, die diesem HTTP-Cookie entspricht.

Domäne

Gibt die Domäne dieses HTTP-Cookies an. Der HTTP-Cookie ist in dieser Domäne gültig.

Pfad

Gibt den Pfad dieses HTTP-Cookies an. Dieser Wert gibt die Teilmenge der URLs an, für die dieser HTTP-Cookie auf dem ursprünglichen HTTP-Server gilt.

ExpirationDate

Gibt das Ablaufdatum dieses HTTP-Cookies an. Einige Cookies geben ein Ablaufdatum zurück, während andere ein maximales Alter zurückgeben. Das Ablaufdatum wird im folgenden Format zurückgegeben:

"yyyy.MM.dd 'um' HH:mm:ss z"

MaxAge

Gibt das maximale Alter dieses HTTP-Cookies an. Einige Cookies geben ein maximales Alter zurück, während andere ein Ablaufdatum zurückgeben.

Kommentar

Gibt den Zweck dieses HTTP-Cookies an.

Ports

Gibt die Ports dieses HTTP-Cookies an. Die Ports werden als Zeichenfolge mit kommagetrennten Werten zurückgegeben. Dieser Wert gibt die Ports an, auf denen dieser HTTP-Cookie in einem Header der Anforderung zurückgeschickt werden kann.

IsSecure

Eine der folgenden Optionen:

- Wahr: Zeigt an, dass dieser HTTP-Cookie nur in einer sicheren Verbindung gesendet werden kann.
- Falsch: Zeigt an, dass für das Senden dieses Cookies keine sichere Verbindung erforderlich.

ResponseHeaderName

Gibt den Namen des Antwort-Headers an, der diesen HTTP-Cookie enthält. Dieser Wert kann "Set-Cookie" oder "Set-Cookie2" sein.

Ergebnis

Dieser Wert ist eine der folgenden Optionen:

1

Zeigt an, dass der Operator erfolgreich abgeschlossen wurde.

-1

Zeigt an, dass der Operator fehlschlagen ist.

Ursache

Dieser Wert ist eine der folgenden Optionen:

Abgeschlossen

Dieser Grund wird dem Ergebnis 1 zugeordnet, erfolgreicher Abschluss.

<error message>

Eine Erklärung zum Auftreten des Fehlers. Dem Ergebnis -1 zugeordnet, wobei der Operator fehlgeschlagen ist.

HTTP-Operatoren: Allgemeine Ausgabe-Ports

Erfolg

Der Operator wurde erfolgreich abgeschlossen.

Zeitlimit

Ein Zeitlimit für Verbindung oder Zeitlimit für Socket wurde erreicht.

Fehlgeschlagen

Die HTTP-Antwort hat einen Statuscode, der größer oder gleich 300 ist.

Der Inhalt der HTTP-Antwort kann den HTTP-Statuscode und den Grund für den Operatorfehler enthalten. "HTTPResponseReasonPhrase" kann einen generischen Grund des Fehlers enthalten. Ein generischer Grund für den Fehler wird in der Statuszeile der HTTP-Antwort zurückgegeben. Für Informationen prüfen Sie "HTTPResponseContent".

Es folgen Beschreibungen für Statuscodes 401 und 407 und andere Fehlergründe:

401

Statuscode 401 zeigt eine der folgenden Bedingungen an:

- Falsche Anmeldeinformationen für die URL-Authentifizierung.
- Falsches Schema für die URL-Authentifizierung (Basis gegen NTLM)
- Es sind keine Anmeldeinformationen für die Authentifizierung angegeben, wenn die HTTP-URL Authentifizierung benötigt.
- Fehler bei der URL-Authentifizierung.

Mit dem Fehlercode 401 gibt der HTTP-Server normalerweise zurück den Antwort-Header *WWW-Authenticate* zurück. Dieser Antwort-Header enthält das Authentifizierungsschema, das der HTTP-Server verwendet. Verwenden Sie diese Informationen, um zu bestimmen, welches Authentifizierungsschema gegen die URL verwendet werden soll. HTTP-Basisauthentifizierung und NTLM-Authentifizierung sind die zwei Schemen, die HTTP-Operatoren unterstützen.

407

Statuscode 407 zeigt eine der folgenden Bedingungen an:

- Falsche Anmeldeinformationen für die Proxy-Authentifizierung.
- Falsches Schema für die Proxy-Authentifizierung (Basis gegen NTLM).
- Es sind keine Anmeldeinformationen für die Authentifizierung angegeben, wenn der Proxy Authentifizierung benötigt.
- Proxy-Authentifizierung schlägt fehl.

Mit dem Fehlercode 407 gibt der HTTP-Proxy normalerweise zurück den Antwort-Header *Proxy-Authenticate* zurück. Dieser Antwort-Header enthält das Authentifizierungsschema, das der Proxy-Server verwendet. Verwenden Sie diese Informationen, um zu bestimmen, welches Authentifizierungsschema gegen den Proxy verwendet werden soll. HTTP-Basisauthentifizierung und NTLM-Authentifizierung sind die zwei Schemen, die HTTP-Operatoren unterstützen.

- Die URL oder der Proxy-Host, die bzw. der als Eingabe angegeben wurde, ist unbekannt.
- Der HTTP-Aufruf geht durch einen Proxy aber die Eingabe enthält keine Spezifikation eines Proxys. In diesem Fall kann der Operator angeben, dass die Verbindung zur HTTP-URL abgelehnt wird.
- Eingabe enthält einen ungültigen Proxy-Port. In diesem Fall kann der Operator angeben, dass die Verbindung zu "ProxyHost:ProxyPort" abgelehnt wird.
- Ungültige Eingabe, wenn z. B. Folgendes entdeckt wurde:
  - Negative Verbindung oder das Socket erreicht ein Zeitlimit.
  - Negative Höchstanzahl von Umleitungen
  - Negative Größenbeschränkung für die Datensatzvariable der Antwort
  - Antwort wird in eine Datei gespeichert, und es wird kein Dateipfad angegeben.

Anwenderspezifische Ports

Wenn während des Prozessdesigns durch den Anwender festgelegt.

Operator "HTTP DELETE"

Der Operator "HTTP DELETE" sendet ein "HTTP DELETE" in eine URL. Der Vorgang "HTTP DELETE" führt dazu, dass der HTTP-Server die Ressource löscht, die sich in der angeforderten URL befindet.

Der Operator "HTTP DELETE" kann für RESTful-Services verwendet werden.

Wichtig! Verwenden Sie den Operator "HTTP OPTIONS", um zu bestimmen, ob die Methode "HTTP DELETE" unterstützt wird. Normalerweise wird die Methode "HTTP DELETE" auf öffentlichen HTTP-Servern deaktiviert, damit Dateien auf den HTTP-Servern nicht gelöscht werden.

Eingabeparameter

In folgenden Abschnitten finden Sie Beschreibungen der Eingabeparameter für den Operator "HTTP DELETE":

Informationen zu HTTP-URL

Beschreibungen der Eingabeparameter finden Sie unter Informationen zu HTTP-URL (siehe Seite 440).

HTTP-Proxy-Informationen

Beschreibungen der Eingabeparameter finden Sie unter HTTP-Proxy-Informationen (siehe Seite 444).

Informationen zu HTTP-Header

Beschreibungen der Eingabeparameter finden Sie unter Informationen zu HTTP-Header (siehe Seite 448).

Informationen zu HTTP-Cookies

Beschreibungen der Eingabeparameter finden Sie unter Informationen zu HTTP-Cookies (siehe Seite 449).

Informationen zum Inhalt der HTTP-Antwort

Beschreibungen der Eingabeparameter finden Sie unter Informationen zum Inhalt der HTTP-Antwort (siehe Seite 449).

Informationen zur HTTP-Konfiguration

Beschreibungen der Eingabeparameter finden Sie unter Informationen zur HTTP-Konfiguration (siehe Seite 451).

Ausgabeparameter

HttpRemoteURL

HttpValidateSSLCert

HttpSvrAuth

HttpSvrNtlmAuth

HttpAuthUser

HttpAuthPwd

HttpAuthDomain

HttpProxy

HttpProxyHost

HttpProxyPort

HttpProxyAuth

HttpProxyNtlmAuth

HttpProxyUser

HttpProxyPwd

HttpProxyDomain

HttpHeaderFieldsType

HttpHeaderFieldsValueMap

HttpHeaderFieldsVarValueMap

HttpCookieFieldsVarValueMap

HttpRespSaveToFile

HttpRespLocalFile

HttpRespLocalFileExists

HttpRespSaveToDataset

HttpRespDatasetVarLimit

HttpVersion

HttpConnectionTimeout

HttpSocketTimeout

HttpHandleRedirects

HttpMaxRedirects

HTTPRequestUrl

HTTPResponseStatusLine

HTTPResponseStatusCode

HTTPResponseReasonPhrase

HTTPResponseProtocolVersion

HTTPResponseContentType

HTTPResponseContentCharset

HTTPResponseContentLength

HTTPResponseContentEncoding

HTTPResponseContentFilePath

HTTPResponseContent

HTTPResponseHtmlContent

HTTPResponseHeaders

HTTPResponseContentIsChunked

HTTPRequestHeaders

HTTPRequestLine

HTTPCookiesStore

Weitere Informationen finden Sie unter HTTP-Operatoren: Allgemeine Ausgabeparameter (siehe Seite 453).

Ausgabe-Ports

Ausgabe-Ports für den Operator "HTTP DELETE" enthalten nur die allgemeinen Ausgabe-Ports (siehe Seite 458) für HTTP-Operatoren.

Operator "HTTP GET"

Der Operator "HTTP GET" sendet eine "HTTP GET"-Anfrage zu einer URL.

Wenn die angegebene URL auf eine Ressource verweist, dann ruft der Operator "HTTP GET" die Ressource ab. Verwenden Sie den Operator "HTTP GET", um eine Datei von einem HTTP-Server herunterzuladen, indem Sie die URL der Datei angeben.

Wenn die angegebene URL auf einen Prozess verweist, der Daten erstellt, dann ruft der Operator "HTTP GET" die vom Prozess erstellten Daten ab. Der Operator "HTTP GET" ruft keine Daten aus der Prozessquelle ab.

Der Operator "HTTP GET" kann für RESTful-Services verwendet werden.

Eingabeparameter

Informationen zu HTTP-URL

Beschreibungen der Eingabeparameter finden Sie unter Informationen zu HTTP-URL (siehe Seite 440).

Informationen zu HTTP GET

URL-Parameter-Kodierung

Gibt die Zeichenkodierung an, die für die Verschlüsselung der URL-Parameter verwendet werden soll. ASCII ist die empfohlene Verschlüsselung für URL-Parameter in einem "HTTP GET"-Betriebsablauf. Die URL-Parameter werden in die URL übertragen. Andere Verschlüsselungen als ASCII, UTF-8 und ISO-8859-1 funktionieren normalerweise nicht.

Werte

Eine der folgenden Komponenten:

- ASCII
- UTF-8
- ISO-8859-1

Standard

Leer - Entspricht dem Wert "ASCII".

Indiziertes ValueMap für URL-Parameter verwenden?

Zeigt an, welche der folgenden Felder für die Eingabe von URL-Parametern verwendet werden soll: "URL-Parameter - Indiziertes ValueMap" oder "URL-Parameter".

Werte

Ausgewählt: Geben Sie die URL-Parameter als indiziertes ValueMap im Feld "URL-Parameter - Indiziertes ValueMap" ein.

Deaktiviert: Geben Sie die URL-Parameter in das Feld "URL-Parameter" ein.

Parameter von URL

Gibt die Namen der URL-Parameter in der Spalte "Schlüssel" und die Werte der URL-Parameter in der Spalte "Wert" an.

Verwenden Sie die Schaltflächen, um Parameter hinzuzufügen, zu entfernen oder neu anzuordnen.

Hinweis: Der Operator ignoriert URL-Parameter, bei denen "Schlüssel" leer ist, das heißt, wenn kein URL-Parametername angegeben ist.

URL-Parameter - Indiziertes ValueMap

Gibt den Namen eines indizierten ValueMap an. Dieser Name ist eine Datensatzvariable eines indizierten ValueMap mit Schlüsseln und Werten, bei denen die Schlüssel URL-Parameternamen sind. Das indizierte ValueMap muss sowohl den Parameter "Schlüssel" als auch den Parameter "Wert" haben. Das indizierte ValueMap muss das gleiche Format haben wie das ValueMap, das im Feld "URL-Parameter" aufgelistet ist. Der Operator ignoriert URL-Parameter mit einem leeren Schlüssel.

HTTP-Proxy-Informationen

Beschreibungen der Eingabeparameter finden Sie unter HTTP-Proxy-Informationen (siehe Seite 444).

Informationen zu HTTP-Header

Beschreibungen der Eingabeparameter finden Sie unter Informationen zu HTTP-Header (siehe Seite 448).

Informationen zu HTTP-Cookies

Beschreibungen der Eingabeparameter finden Sie unter Informationen zu HTTP-Cookies (siehe Seite 449).

Informationen zum Inhalt der HTTP-Antwort

Beschreibungen der Eingabeparameter finden Sie unter Informationen zum Inhalt der HTTP-Antwort (siehe Seite 449).

Informationen zur HTTP-Konfiguration

Beschreibungen der Eingabeparameter finden Sie unter Informationen zur HTTP-Konfiguration (siehe Seite 451).

Ausgabeparameter

HttpRemoteURL

HttpValidateSSLCert

HttpSvrAuth

HttpSvrNtlmAuth

HttpAuthUser

HttpAuthPwd

HttpAuthDomain

HttpReqUrlParamsEncoding

HttpReqUrlParamsType

HttpReqUrlParamsValueMap

HttpReqUrlParamsVarValueMap

HttpProxy

HttpProxyHost

HttpProxyPort

HttpProxyAuth

HttpProxyNtlmAuth

HttpProxyUser

HttpProxyPwd

HttpProxyDomain

HttpHeaderFieldsType

HttpHeaderFieldsValueMap

HttpHeaderFieldsVarValueMap

HttpCookieFieldsVarValueMap

HttpRespSaveToFile

HttpRespLocalFile

HttpRespLocalFileExists

HttpRespSaveToDataset

HttpRespDatasetVarLimit

HttpVersion

HttpConnectionTimeout

HttpSocketTimeout

HttpHandleRedirects

HttpMaxRedirects

HTTPRequestUrl

HTTPResponseStatusLine

HTTPResponseStatusCode

HTTPResponseReasonPhrase

HTTPResponseProtocolVersion

HTTPResponseContentType

HTTPResponseContentCharset

HTTPResponseContentLength

HTTPResponseContentEncoding

HTTPResponseContentFilePath

HTTPResponseContent

HTTPResponseHtmlContent

HTTPResponseHeaders

TTPResponseContentIsChunked

HTTPRequestHeaders

HTTPRequestLine

HTTPCookiesStore

Weitere Informationen finden Sie unter HTTP-Operatoren: Allgemeine Ausgabeparameter (siehe Seite 453).

Ausgabe-Ports

Ausgabe-Ports für den Operator "HTTP GET" enthalten allgemeine Ausgabe-Ports für HTTP-Operatoren und einen zusätzlichen Fehlerfall.

Operator-Ausfall

Der Operator "HTTP GET" kann aus folgenden Gründen fehlschlagen:

- Allgemeine Fehler der Ausgangs-Ports für alle HTTP-Operatoren.

  Hinweis: Beschreibungen finden Sie unter HTTP-Operatoren: Allgemeine Ausgabe-Ports (siehe Seite 458).

- Sie haben eine ungültige Kodierung im Feld "URL-Parameter-Kodierung" angegeben.

Operator "HTTP HEAD"

Der Operator "HTTP HEAD" sendet eine "HTTP HEAD"-Anfrage zu einer URL. Die Methode "HTTP HEAD" ähnelt der Methode "HTTP GET". Der Unterschied zwischen den beiden Methoden ist, dass der HTTP-Server mit "HTTP HEAD" die Ressource, die sich unter der URL befindet, nicht zurückgibt. Die HTTP-HEADER der Antwort sind identisch für die Methode "HEAD" und "GET".

Die Methode "HTTP HEAD" wird normalerweise verwendet, um Informationen über eine Ressource zu erhalten, ohne sie tatsächlich abzurufen (zu übertragen).

Eingabeparameter

Informationen zu HTTP-URL

Beschreibungen der Eingabeparameter finden Sie unter Informationen zu HTTP-URL (siehe Seite 440).

Informationen zu HTTP HEAD

URL-Parameter-Kodierung

Gibt die Zeichenkodierung an, die für die Verschlüsselung der URL-Parameter verwendet werden soll. ASCII ist die empfohlene Verschlüsselung für URL-Parameter in einem "HTTP HEAD"-Betriebsablauf. Die URL-Parameter werden in die URL übertragen. Andere Verschlüsselungen als ASCII, UTF-8 und ISO-8859-1 funktionieren normalerweise nicht.

Werte

Eine der folgenden Komponenten:

- ASCII
- UTF-8
- ISO-8859-1

Standard

Leer - Entspricht dem Wert "ASCII".

Indiziertes ValueMap für URL-Parameter verwenden?

Zeigt an, welche der folgenden Felder für die Eingabe von URL-Parametern verwendet werden soll. Zeigt vor allem an, ob URL-Parameter als indiziertes ValueMap im Feld "URL-Parameter - Indiziertes ValueMap" angegeben werden sollen, oder ob die URL-Parameter in das Feld "URL-Parameter" eingegeben werden sollen.

Werte

Ausgewählt: Geben Sie die URL-Parameter als indiziertes ValueMap im Feld "URL-Parameter - Indiziertes ValueMap" ein.

Deaktiviert: Geben Sie die URL-Parameter in das Feld "URL-Parameter" ein.

Parameter von URL

Gibt die Namen der URL-Parameter in der Spalte "Schlüssel" und die Werte der URL-Parameter in der Spalte "Wert" an.

Verwenden Sie die Schaltflächen, um Parameter hinzuzufügen, zu entfernen oder neu anzuordnen.

Hinweis: Der Operator ignoriert URL-Parameter, bei denen "Schlüssel" leer ist, das heißt, wenn kein URL-Parametername angegeben ist.

URL-Parameter - Indiziertes ValueMap

Gibt den Namen eines indizierten ValueMap an. Dieser Name ist eine Datensatzvariable eines indizierten ValueMap mit Schlüsseln und Werten, bei denen die Schlüssel URL-Parameternamen sind. Das indizierte ValueMap muss sowohl den Parameter "Schlüssel" als auch den Parameter "Wert" haben. Das indizierte ValueMap muss das gleiche Format haben wie das ValueMap, das im Feld "URL-Parameter" aufgelistet ist. Der Operator ignoriert URL-Parameter mit einem leeren Schlüssel.

HTTP-Proxy-Informationen

Beschreibungen der Eingabeparameter finden Sie unter HTTP-Proxy-Informationen (siehe Seite 444).

Informationen zu HTTP-Header

Beschreibungen der Eingabeparameter finden Sie unter Informationen zu HTTP-Header (siehe Seite 448).

Informationen zu HTTP-Cookies

Beschreibungen der Eingabeparameter finden Sie unter Informationen zu HTTP-Cookies (siehe Seite 449).

Informationen zum Inhalt der HTTP-Antwort

Beschreibungen der Eingabeparameter finden Sie unter Informationen zum Inhalt der HTTP-Antwort (siehe Seite 449).

Informationen zur HTTP-Konfiguration

Beschreibungen der Eingabeparameter finden Sie unter Informationen zur HTTP-Konfiguration (siehe Seite 451).

Ausgabeparameter

HttpRemoteURL

HttpValidateSSLCert

HttpSvrAuth

HttpSvrNtlmAuth

HttpAuthUser

HttpAuthPwd

HttpAuthDomain

HttpReqUrlParamsEncoding

HttpReqUrlParamsType

HttpReqUrlParamsValueMap

HttpReqUrlParamsVarValueMap

HttpProxy

HttpProxyHost

HttpProxyPort

HttpProxyAuth

HttpProxyNtlmAuth

HttpProxyUser

HttpProxyPwd

HttpProxyDomain

HttpHeaderFieldsType

HttpHeaderFieldsValueMap

HttpHeaderFieldsVarValueMap

HttpCookieFieldsVarValueMap

HttpRespSaveToFile

HttpRespLocalFile

HttpRespLocalFileExists

HttpRespSaveToDataset

HttpRespDatasetVarLimit

HttpVersion

HttpConnectionTimeout

HttpSocketTimeout

HttpHandleRedirects

HttpMaxRedirects

HTTPRequestUrl

HTTPResponseStatusLine

HTTPResponseStatusCode

HTTPResponseReasonPhrase

HTTPResponseProtocolVersion

HTTPResponseContentType

HTTPResponseContentCharset

HTTPResponseContentLength

HTTPResponseContentEncoding

HTTPResponseContentFilePath

HTTPResponseContent

HTTPResponseHtmlContent

HTTPResponseHeaders

HTTPResponseContentIsChunked

HTTPRequestHeaders

HTTPRequestLine

HTTPCookiesStore

Weitere Informationen finden Sie unter HTTP-Operatoren: Allgemeine Ausgabeparameter (siehe Seite 453).

Hinweis: Der Operator "HTTP HEAD" überträgt nicht die Ressource, die sich in der angegebenen URL befindet. Deswegen sind Ausgabevariablen wie "HTTPResponseContentType" und "HTTPResponseContentLength" leer. Um Informationen über die Ressource in der angegebenen URL zu finden, zeigen Sie "HTTPResponseHeaders" an. "HTTPResponseHeaders" enthält Informationen zu Headers wie z. B. Inhaltstyp und Inhaltslänge. Dies sind Informationen, die über den HTTP-Server auf diese Ressource zurückgegeben werden.

Ausgabe-Ports

Ausgabe-Ports für den Operator "HTTP HEAD" enthalten allgemeine Ausgabe-Ports für HTTP-Operatoren und einen zusätzlichen Fehlerfall.

Operator-Ausfall

Der Operator "HTTP HEAD" kann aus folgenden Gründen fehlschlagen:

- Allgemeine Fehler der Ausgangs-Ports für alle HTTP-Operatoren.

  Hinweis: Beschreibungen finden Sie unter HTTP-Operatoren: Allgemeine Ausgabe-Ports (siehe Seite 458).
- Der Anwender hat eine ungültige Verschlüsselung im Feld "URL-Parameter-Kodierung" angegeben.

Operator "HTTP OPTIONS"

Der Operator "HTTP OPTIONS" sendet eine "HTTP OPTIONS"-Anfrage zu einer URL. Die HTTP-Optionen ermöglichen es Ihnen, zu bestimmen, welche HTTP-Methoden der HTTP-Server unterstützt.

Die unterstützten HTTP-Methoden werden am Ende der Ausführung des Operators in der Ausgabevariablen "HTTPAllowedMethods" aufgelistet.

Eingabeparameter

In folgenden Abschnitten finden Sie Beschreibungen der Eingabeparameter für den Operator "HTTP OPTIONS":

Informationen zu HTTP-URL

Beschreibungen der Eingabeparameter finden Sie unter Informationen zu HTTP-URL (siehe Seite 440).

HTTP-Proxy-Informationen

Beschreibungen der Eingabeparameter finden Sie unter HTTP-Proxy-Informationen (siehe Seite 444).

Informationen zu HTTP-Header

Beschreibungen der Eingabeparameter finden Sie unter Informationen zu HTTP-Header (siehe Seite 448).

Informationen zu HTTP-Cookies

Beschreibungen der Eingabeparameter finden Sie unter Informationen zu HTTP-Cookies (siehe Seite 449).

Informationen zum Inhalt der HTTP-Antwort

Beschreibungen der Eingabeparameter finden Sie unter Informationen zum Inhalt der HTTP-Antwort (siehe Seite 449).

Informationen zur HTTP-Konfiguration

Beschreibungen der Eingabeparameter finden Sie unter Informationen zur HTTP-Konfiguration (siehe Seite 451).

Ausgabeparameter

HttpRemoteURL

HttpValidateSSLCert

HttpSvrAuth

HttpSvrNtlmAuth

HttpAuthUser

HttpAuthPwd

HttpAuthDomain

HttpProxy

HttpProxyHost

HttpProxyPort

HttpProxyAuth

HttpProxyNtlmAuth

HttpProxyUser

HttpProxyPwd

HttpProxyDomain

HttpHeaderFieldsType

HttpHeaderFieldsValueMap

HttpHeaderFieldsVarValueMap

HttpCookieFieldsVarValueMap

HttpRespSaveToFile

HttpRespLocalFile

HttpRespLocalFileExists

HttpRespSaveToDataset

HttpRespDatasetVarLimit

HttpVersion

HttpConnectionTimeout

HttpSocketTimeout

HttpHandleRedirects

HttpMaxRedirects

HTTPRequestUrl

HTTPResponseStatusLine

HTTPResponseStatusCode

HTTPResponseReasonPhrase

HTTPResponseProtocolVersion

HTTPResponseContentType

HTTPResponseContentCharset

HTTPResponseContentLength

HTTPResponseContentEncoding

HTTPResponseContentFilePath

HTTPResponseContent

HTTPResponseHtmlContent

HTTPResponseHeaders

HTTPResponseContentIsChunked

HTTPRequestHeaders

HTTPRequestLine

HTTPCookiesStore

HTTPAllowedMethods

Gibt eine indizierte Zeichenfolge an, die die Namen der Methoden enthält, die von der Ressource oder dem Server unterstützt werden, der sich in der angegebenen URL befindet.

Weitere Informationen finden Sie unter HTTP-Operatoren: Allgemeine Ausgabeparameter (siehe Seite 453).

Ausgabe-Ports

Ausgabe-Ports für den Operator "HTTP OPTIONS" enthalten nur die allgemeinen Ausgabe-Ports für HTTP-Operatoren.

Beschreibungen finden Sie unter HTTP-Operatoren: Allgemeine Ausgabe-Ports (siehe Seite 458).

Operator "HTTP POST"

Der Operator "HTTP POST" sendet eine "HTTP POST"-Anfrage zu einer URL. Der Operator "HTTP POST" fordert normalerweise an, dass der HTTP-Server die Ressource speichert, die als Inhalt der HTTP-Anforderung eingeschlossen ist. Der HTTP-Serverprozess unter der angegebenen URL bearbeitet dann die Ressource.

Hinweis: Im Gegensatz zum Operator "HTTP PUT", verweist die URL eines Operator "HTTP POST" auf den Prozess, der die eingeschlossene Ressource verarbeiten kann.

Der Operator "HTTP POST" kann für RESTful-Services verwendet werden.

Eingabeparameter

Informationen zu HTTP-URL

Beschreibungen der Eingabeparameter finden Sie unter Informationen zu HTTP-URL (siehe Seite 440).

Informationen zu HTTP POST

"Informationen zu HTTP POST" gibt den Text der HTTP-Anforderung an.

Aufgeteilt?

Gibt an, ob die HTTP-Anforderung geteilt gesendet werden soll.

Wenn eine Blockkodierung festgelegt ist, enthält die HTTP-Anforderung nicht den Header "Inhaltslänge".

Hinweis: HTTP 1.0 unterstützt keine Blockkodierung. Der Operator "HTTP POST" schlägt mit einem "Fehler im HTTP-Client-Protokoll" fehl, wenn Blockkodierung in einer HTTP-Version 1.0 festgelegt ist.

Werte

Eine der folgenden Komponenten:

- Wahr: Zeigt an, dass die HTTP-Anforderung aufgeteilt gesendet werden soll.
- Falsch: Zeigt an, dass die HTTP-Anforderung nicht aufgeteilt gesendet werden soll.
- Anderer Wert: Entspricht dem Wert "Falsch".

Standard

Leer - Entspricht dem Wert "Falsch".

Inhaltstyp

Gibt den Typ des Textinhalts der HTTP-Anforderung an, der in der HTTP-Anforderung als Header (Inhaltstyp) festgelegt ist.

Werte

Dieser Wert kann einer der folgenden sein:

- Ein aus der Drop-down-Liste ausgewählter Medientyp.
- Ein leerer Wert, wenn der Inhalt aus einer Datei abgerufen wird, die in "Pfad der Inhaltsdatei" angegeben ist.

  CA Process Automation legt den Wert auf *application/octet-stream* fest. Der HTTP-Server ist für die Interpretation dieses generischen Inhaltstyps verantwortlich.
- Ein leerer Wert, wenn der Inhalt aus dem Feld "Inhalt" abgerufen wird.

  CA Process Automation legt den Inhaltstyp nicht fest. Der HTTP-Server ist für die Interpretation des Headers "No Content-Type" verantwortlich.
- Ein gültiger Medientyp, den Sie in das Feld eingeben.

  Gültige Medientypen finden Sie auf den Seiten der IANA-Website unter "Assignments/media types".

Hinweis: Stellen Sie sicher, dass Sie den richtigen "Content-Type" festlegen, vor allem, wenn der Inhalt nicht aus einer Datei abgerufen wird.

Zeichenkodierung für Inhalt

Gibt die Zeichenkodierung für den Inhalt des Textes der HTTP-Anforderung an. Legen Sie dieses Feld nur dann fest, wenn der Inhaltstyp nur Zeichen umfasst, beispielsweise "Text/XXX".

Werte

Dieser Wert kann einer der folgenden sein:

- Ein aus der Drop-down-Liste ausgewählter Zeichensatz.
- Ein gültiger Zeichensatz (Kodierung), den Sie in das Feld eingeben.

  Gültige Kodierungen finden Sie auf den Seiten der IANA-Website unter "Assignments/character-sets".

Hinweis: Stellen Sie sicher, dass Sie die richtige Zeichenkodierung festlegen, vor allem, wenn der Inhalt nicht aus einer Datei abgerufen wird.

Inhalt aus Datei abrufen?

Gibt an, ob der Text der HTTP-Anforderung aus einer lokalen Datei auf dem Host, auf dem der Kontaktpunkt ausgeführt wird, abgerufen werden soll.

Werte

Dieser Wert kann einer der folgenden sein:

- Ausgewählt: Zeigt an, dass der Text der HTTP-Anforderung aus einer lokalen Datei auf dem Host, auf dem der Kontaktpunkt ausgeführt wird, abgerufen werden soll.
- Deaktiviert: Zeigt an, dass der Text der HTTP-Anforderung aus dem Feld "Inhalt" abgerufen werden soll.

Pfad der Inhaltsdatei

Gibt den Pfad zu der lokalen Datei auf dem Host ein, auf dem der Kontaktpunkt ausgeführt wird. Die lokale Datei enthält den Text der HTTP-Anforderung.

Inhalt

Gibt den Text der HTTP-Anforderung an.

HTTP-Proxy-Informationen

Beschreibungen der Eingabeparameter finden Sie unter HTTP-Proxy-Informationen (siehe Seite 444).

Informationen zu HTTP-Header

Beschreibungen der Eingabeparameter finden Sie unter Informationen zu HTTP-Header (siehe Seite 448).

Informationen zu HTTP-Cookies

Beschreibungen der Eingabeparameter finden Sie unter Informationen zu HTTP-Cookies (siehe Seite 449).

Informationen zum Inhalt der HTTP-Antwort

Beschreibungen der Eingabeparameter finden Sie unter Informationen zum Inhalt der HTTP-Antwort (siehe Seite 449).

Informationen zur HTTP-Konfiguration

Beschreibungen der Eingabeparameter finden Sie unter Informationen zur HTTP-Konfiguration (siehe Seite 451).

Ausgabeparameter

HttpRemoteURL

HttpValidateSSLCert

HttpSvrAuth

HttpSvrNtlmAuth

HttpAuthUser

HttpAuthPwd

HttpAuthDomain

HttpReqIsChunked

HttpReqContentType

HttpReqContentCharset

HttpReqContentFromFile

HttpReqContentFilePath

HttpReqContent

HttpProxy

HttpProxyHost

HttpProxyPort

HttpProxyAuth

HttpProxyNtlmAuth

HttpProxyUser

HttpProxyPwd

HttpProxyDomain

HttpHeaderFieldsType

HttpHeaderFieldsValueMap

HttpHeaderFieldsVarValueMap

HttpCookieFieldsVarValueMap

HttpRespSaveToFile

HttpRespLocalFile

HttpRespLocalFileExists

HttpRespSaveToDataset

HttpRespDatasetVarLimit

HttpVersion

HttpConnectionTimeout

HttpSocketTimeout

HttpHandleRedirects

HttpMaxRedirects

HTTPRequestUrl

HTTPResponseStatusLine

HTTPResponseStatusCode

HTTPResponseReasonPhrase

HTTPResponseProtocolVersion

HTTPResponseContentType

HTTPResponseContentCharset

HTTPResponseContentLength

HTTPResponseContentEncoding

HTTPResponseContentFilePath

HTTPResponseContent

HTTPResponseHtmlContent

HTTPResponseHeaders

HTTPResponseContentIsChunked

HTTPRequestHeaders

HTTPRequestLine

HTTPCookiesStore

Weitere Informationen finden Sie unter HTTP-Operatoren: Allgemeine Ausgabeparameter (siehe Seite 453).

Ausgabe-Ports

Ausgabe-Ports für den Operator "HTTP POST" enthalten allgemeine Ausgabe-Ports für HTTP-Operatoren und einen zusätzlichen Fehlerfall.

Operator-Ausfall

Der Operator "HTTP POST" kann aus folgenden Gründen fehlschlagen:

- Allgemeine Fehler der Ausgangs-Ports für alle HTTP-Operatoren.

  Hinweis: Beschreibungen finden Sie unter HTTP-Operatoren: Allgemeine Ausgabe-Ports (siehe Seite 458).

- Es kann ein Fehler im HTTP-Client-Protokoll auftreten, wenn in der HTTP-Version 1.0 Blockkodierung festgelegt ist. Dies kann zu "ClientProtocolException" führen.
- Die Eingabe enthält eine ungültige Kodierung im Feld "Zeichenkodierung für Inhalt".

  Hinweis: CA Process Automation überprüft nicht, ob der angegebene Wert "Inhaltstyp" in der Eingabe gültig ist, wenn der Inhalt aus einer Datei abgerufen wurde. Der Empfänger-HTTP-Server muss einen Fehlercode zurückgeben, wenn ein ungültiger Inhaltstyp entdeckt wird.

Operator "HTTP POST FORM"

Der Operator "HTTP POST FORM" sendet Daten an ein HTTP-Formular. Die Daten entsprechen dem Typ "Name-Wert-Paare", die Werte enthalten können, die Dateien sind.

Im Gegensatz zum Operator "HTTP POST" sendet der Operator "HTTP POST FORM" die "HTTP POST"-Anfrage an ein HTTP-Formular. Die URL des Operators ist das Aktionselement des Formulars. Beispiel:

```
<form name=”myForm” method=”post” action =”XXXXXXXX”>
```

Drücken Sie die Ziel-URL als vollständigen Pfad zum Skript oder als Prozess auf dem HTTP-Server aus, auf den das Aktionselement verweist. Verwenden Sie keinen relativen Pfad für die Ziel-URL.

Das Skript oder der Prozess, auf den das Aktionselement verweist, muss *öffentlich* für den Abruf durch den Operator "HTTP POST FORM" verfügbar sein.

Eingabeparameter

Informationen zu HTTP-URL

Beschreibungen der Eingabeparameter finden Sie unter Informationen zu HTTP-URL (siehe Seite 440).

Informationen zu HTTP POST FORM

Kodierung für Formularfelder

Gibt die Zeichenkodierung an, die für die Kodierung der Parameter der Formularfelder verwendet werden soll. Die Daten im Operator "HTTP POST FORM" gehen in den Text der Anfrage.

Werte

Dieses Feld akzeptiert folgende Werte:

- Einen Wert aus der Drop-down-Liste.
- Einen vom Anwender eingegebenen Wert.
- Leerer Wert: ISO-8859-1

Standard

ISO-8859-1

Indiziertes ValueMap für Formularfelder verwenden?

Gibt an, ob Formularfelder im Feld "Formularfelder - Indiziertes ValueMap" oder im Feld "Formularfelder" angegeben werden sollen. Die Auswahl aktiviert die Eingabe für das entsprechende Feld.

Werte

- Ausgewählt: Zeigt an, dass die Formularfelder als indiziertes ValueMap im Feld "Formularfelder - Indiziertes ValueMap" angegeben sind.
- Deaktiviert: Zeigt an, dass die Formularfelder im Feld "Formularfelder" angegeben sind.

Formularfelder

Gibt die Kombination aus Schlüssel und Wert für jedes Formularfeld an. Die Spalte "Schlüssel" enthält die Namen der Formularfelder, und die Spalte "Wert" enthält die Werte der Formularfelder.

Verwenden Sie die Schaltflächen, um Parameter hinzuzufügen, zu entfernen oder neu anzuordnen.

Hinweis: Der Operator "HTTP POST FORM" ignoriert ein Formularfeld mit einem leeren Schlüssel.

Formularfelder - Indiziertes ValueMap

Gibt den Namen eines indizierten ValueMap an.

Dieses ValueMap besteht aus Schlüssel und Wert. Eine Eingabe in "Schlüssel" ist der Name eines Formularfelds. Die entsprechende Eingabe in "Wert" ist der Wert dieses Formularfelds. Das indizierte ValueMap hat das gleiche Format wie das ValueMap, das im Feld "Formularfelder" aufgelistet ist.

Hinweis: Der Operator "HTTP POST FORM" ignoriert jedes Formularfeld mit einem leeren Schlüssel.

Dateifelder in Formular?

Zeigt an, ob das Formular Felder enthält, mit denen Dateien in das Formular hochgeladen werden können. Wenn Sie dieses Kontrollkästchen auswählen, können Sie zugehörige Daten in die folgenden Felder eingeben: "Indiziertes ValueMap für Formulardateien verwenden?", "Formulardateien" und "Formulardateien - Indiziertes ValueMap".

Hinweis: Die HTTP-Anforderungs-Header "Content-Type" hängt von der Einstellung des Felds "Dateifelder in Formular?" ab.

Werte

Dieser Status kann folgendermaßen festgelegt werden:

- Ausgewählt: Der Inhaltstyp der HTTP-Anforderung folgt, wo XXXXXXX die Grenzzeichenfolge ist, die die verschiedenen Teile der HTTP-Anforderung trennt. Jeder Teil im Text der HTTP-Anforderung kann über seinen eigenen Inhaltstyp verfügen.

  ```
  content-type=multipart/form-data;boundary=XXXXXXX
  ```

  - Die *Formularfelder* werden mit der Kodierung verschlüsselt, die im Feld "Kodierung für Formularfelder" angegeben ist. Jedes Formularfeld wird als separater Teil im Text der Anfrage platziert.
  - Jeder Teil des Formularfelds verfügt über content-type=text/plain; charset='encoding', wobei 'encoding' der Wert (oder Standardwert) von "Kodierung für Formularfelder".
  - Die *Formulardateien* werden mit der Kodierung verschlüsselt, die in den Spalten "ContentType" und "ContentCharacterEncoding" (sofern zutreffend), die der einzelnen Formulardatei zugeordnet sind, angegeben ist. Jede Formulardatei wird als separater Teil im Text der Anfrage platziert.
  - Der Content-Type von jeder Formulardatei ist in der Spalte "ContentType" von "Formulardateien" oder "Formulardateien - Indiziertes ValueMap" angegeben. Sofern zutreffend wird diese Spezifikation mit "ContentCharacterEncoding" verbunden.

- Deaktiviert: Der Inhaltstyp der HTTP-Anforderung folgt, wo 'encoding' der Kodierungstyp ist, der in "Kodierung für Formularfelder" angegeben ist.

  ```
  content-type=application/x-www-form-urlencoded;charset='encoding'
  ```

  - Die Formularfelder sind URL-kodiert (mithilfe von 'encoding').
  - Die Formularfelder werden im Text der Anfrage platziert.

Indiziertes ValueMap für Formulardateien verwenden?

Zeigt an, ob die Formulardateifelder im Feld "Formulardateien" oder im Feld "Formulardateien - Indiziertes ValueMap" angegeben werden sollen.

- Ausgewählt: Zeigt an, dass die Formulardateifelder als indiziertes ValueMap im Feld "Formulardateien - Indiziertes ValueMap" angegeben sind.
- Deaktiviert: Zeigt an, dass die Formulardateifelder im Feld "Formularfelder" angegeben sind.

Formulardateien

Gibt jede Formulardatei mit vier Feldern an: "Schlüssel", "Dateipfad", "Inhaltstyp" und "Zeichenkodierung für Inhalt". Es folgen Beschreibungen von allen Feldern einer Formulardatei:

Schlüssel

Gibt die Namen der Formulardateifelder an.

Dateipfad

Gibt den Pfad zu der Datei an, die hochgeladen werden soll. (Dieses Feld ist erforderlich.) Der Operator ignoriert jedes Formulardateifeld mit einem leeren Wert in "FilePath".

Inhaltstyp

Gibt den Inhaltstyp der Datei an, die hochgeladen werden soll. Gültige Inhaltstypen werden auf der IANA-Website unter "Assignments/media-types" aufgelistet. Wenn dieser Wert leer ist, wird der Inhaltstyp vom Teil der entsprechenden Datei im Text der HTTP-Anforderung auf "application/octet-stream" festgelegt. Der HTTP-Server muss dann diesen generischen Inhaltstyp interpretieren.

ContentCharacterEncoding

Gibt den Zeichensatz des Inhalts der Dateien an, die hochgeladen werden sollen, wenn der entsprechende Inhaltstyp nur Zeichen umfasst, beispielsweise "Text/XXX". Lassen Sie für andere Arten von Inhaltstypen diesen Wert leer.

Hinweis: Eine Liste der verschiedenen Zeichensätze (Kodierungen) finden Sie auf den Seiten der IANA-Website unter "Assignments/character-sets".

Formulardateien - Indiziertes ValueMap

Gibt den Namen eines indizierten ValueMap an, das die Namen und die entsprechenden Werte der Formulardateifelder enthält. Das indizierte ValueMap muss das gleiche Format haben wie das ValueMap, das im Feld "Formulardateien" aufgelistet ist. Das heißt, es muss aus "Schlüssel", "FilePath", "ContentType" und "ContentCharacterEncoding" bestehen.

HTTP-Proxy-Informationen

Beschreibungen der Eingabeparameter finden Sie unter HTTP-Proxy-Informationen.

Informationen zu HTTP-Header

Beschreibungen der Eingabeparameter finden Sie unter Informationen zu HTTP-Header (siehe Seite 448).

Informationen zu HTTP-Cookies

Beschreibungen der Eingabeparameter finden Sie unter Informationen zu HTTP-Cookies (siehe Seite 449).

Informationen zum Inhalt der HTTP-Antwort

Beschreibungen der Eingabeparameter finden Sie unter Informationen zum Inhalt der HTTP-Antwort (siehe Seite 449).

Informationen zur HTTP-Konfiguration

Beschreibungen der Eingabeparameter finden Sie unter Informationen zur HTTP-Konfiguration (siehe Seite 451).

Ausgabeparameter

HttpRemoteURL

HttpValidateSSLCert

HttpSvrAuth

HttpSvrNtlmAuth

HttpAuthUser

HttpAuthPwd

HttpAuthDomain

HttpFormFieldsEncoding

HttpFormFieldsType

HttpFormFieldsValueMap

HttpFormFieldsVarValueMap

HttpFormMultipartPost

HttpFormFilesToMultipartType

HttpFormFilesToMultipartValueMap

HttpFormFilesToMultipartVarValueMap

HttpProxy

HttpProxyHost

HttpProxyPort

HttpProxyAuth

HttpProxyNtlmAuth

HttpProxyUser

HttpProxyPwd

HttpProxyDomain

HttpHeaderFieldsType

HttpHeaderFieldsValueMap

HttpHeaderFieldsVarValueMap

HttpCookieFieldsVarValueMap

HttpRespSaveToFile

HttpRespLocalFile

HttpRespLocalFileExists

HttpRespSaveToDataset

HttpRespDatasetVarLimit

HttpVersion

HttpConnectionTimeout

HttpSocketTimeout

HttpHandleRedirects

HttpMaxRedirects

HTTPRequestUrl

HTTPResponseStatusLine

HTTPResponseStatusCode

HTTPResponseReasonPhrase

HTTPResponseProtocolVersion

HTTPResponseContentType

HTTPResponseContentCharset

HTTPResponseContentLength

HTTPResponseContentEncoding

HTTPResponseContentFilePath

HTTPResponseContent

HTTPResponseHtmlContent

HTTPResponseHeaders

HTTPResponseContentIsChunked

HTTPRequestHeaders

HTTPRequestLine

HTTPCookiesStore

Weitere Informationen finden Sie unter HTTP-Operatoren: Allgemeine Ausgabeparameter (siehe Seite 453).

Ausgabe-Ports

Ausgabe-Ports für den Operator "HTTP POST" enthalten allgemeine Ausgabe-Ports für HTTP-Operatoren und einen zusätzlichen Fehlerfall.

Operator-Ausfall

Der Operator "HTTP POST FORM" kann aus folgenden Gründen fehlschlagen:

- Allgemeine Fehler der Ausgangs-Ports für alle HTTP-Operatoren.

  Hinweis: Beschreibungen finden Sie unter HTTP-Operatoren: Allgemeine Ausgabe-Ports (siehe Seite 458).

- Die Eingabe enthält eine ungültige Kodierung in den Feldern "Kodierung für Formularfelder" oder "ContentCharacterEncoding".

  Hinweis: CA Process Automation überprüft nicht, ob "Inhaltstyp" in der Eingabe gültig ist, wenn der Inhalt aus einer Datei abgerufen wurde. Der Empfänger-HTTP-Server muss einen Fehlercode zurückgeben, wenn ein ungültiger Inhaltstyp entdeckt wird.

Operator "HTTP PUT"

Der Operator "HTTP PUT" sendet eine "HTTP PUT"-Anfrage zu einer URL. Der Operator "HTTP PUT" fordert an, dass die Ressource, die als Inhalt der HTTP-Anforderung eingeschlossen ist, unter der angegebenen URL auf dem HTTP-Server gespeichert wird. Die URL muss es CA Process Automation ermöglichen, eine Ressource zu erstellen oder eine vorhandene Ressource zu ersetzen.

Wenn die URL auf eine vorhandene Ressource verweist, bearbeitet der HTTP-Server die eingeschlossene Ressource wie eine geänderte Version der vorhandenen Ressource.

Wenn die URL auf keine vorhandene Ressource verweist, dann erstellt der HTTP-Server eine Ressource mit dem Inhalt der HTTP-Anforderung. Der HTTP-Server speichert dann die neue Ressource unter der angegebenen URL.

Hinweis: Im Gegensatz zum Operator "HTTP POST" identifiziert die URL eines "HTTP PUT"-Operators die Ressource, die im Inhalt der HTTP-Anforderung eingeschlossen ist.

Der Operator "HTTP PUT" kann für RESTful-Services verwendet werden.

Wichtig! Verwenden Sie den Operator "HTTP OPTIONS", um zu bestimmen, ob Sie den Operator "HTTP PUT" verwenden können. Die Methode "HTTP PUT" ist normalerweise auf öffentlichen HTTP-Servern deaktiviert.

Eingabeparameter

Informationen zu HTTP-URL

Beschreibungen der Eingabeparameter finden Sie unter Informationen zu HTTP-URL (siehe Seite 440).

Informationen zu HTTP PUT

"Informationen zu HTTP PUT" gibt den Text der HTTP-Anforderung an.

Aufgeteilt?

Gibt an, ob die HTTP-Anforderung geteilt gesendet werden soll.

Wenn eine Blockkodierung festgelegt ist, enthält die HTTP-Anforderung nicht den Header "Inhaltslänge".

Hinweis: HTTP 1.0 unterstützt keine Blockkodierung. Der Operator "HTTP PUT" schlägt mit einem "Fehler im HTTP-Client-Protokoll" fehl, wenn Blockkodierung in einer HTTP-Version 1.0 festgelegt ist.

Werte

- Wahr: Zeigt an, dass die HTTP-Anforderung aufgeteilt gesendet werden soll.
- Falsch: Zeigt an, dass die HTTP-Anforderung nicht aufgeteilt gesendet werden soll.
- Anderer Wert: Entspricht dem Wert "Falsch".

Standard

Leer - Entspricht dem Wert "Falsch".

Inhaltstyp

Gibt den Typ des Textinhalts der HTTP-Anforderung an, der in der HTTP-Anforderung als Header (Inhaltstyp) festgelegt ist.

Werte

Dieser Wert kann einer der folgenden sein:

- Ein Wert, der aus der Drop-down-Liste von verschiedenen Medientypen ausgewählt wird.
- Ein leerer Wert, wenn der Inhalt aus einer Datei abgerufen wird, die in "Pfad der Inhaltsdatei" angegeben ist.

  CA Process Automation legt den Wert auf *application/octet-stream* fest. Der HTTP-Server ist dann für die Interpretation dieses generischen Inhaltstyps verantwortlich.

- Ein leerer Wert, wenn der Inhalt aus dem Feld "Inhalt" abgerufen wird.

  CA Process Automation legt den Inhaltstyp nicht fest. Der HTTP-Server ist dann für die Interpretation des Headers "No Content-Type" verantwortlich.

- Ein gültiger Medientyp, den Sie manuell in das Feld eingeben.

  Gültige Medientypen finden Sie auf den Seiten der IANA-Website unter "Assignments/media types".

Hinweis: Stellen Sie sicher, dass Sie den richtigen "Content-Type" festlegen, vor allem, wenn der Inhalt nicht aus einer Datei abgerufen wird.

Zeichenkodierung für Inhalt

Gibt die Zeichenkodierung für den Inhalt des Textes der HTTP-Anforderung an. Legen Sie dieses Feld nur dann fest, wenn der Inhaltstyp nur Zeichen umfasst, beispielsweise "Text/XXX".

Werte

Dieser Wert kann einer der folgenden sein:

- Ein Wert, der aus der Drop-down-Liste von verschiedenen Zeichensätzen (Kodierungen) ausgewählt wird.
- Ein gültiger Zeichensatz (Kodierung), den Sie manuell in das Feld eingeben.

  Gültige Kodierungen finden Sie auf den Seiten der IANA-Website unter "assignments/character-sets".

Hinweis: Stellen Sie sicher, dass Sie die richtige Zeichenkodierung festlegen, vor allem, wenn der Inhalt nicht aus einer Datei abgerufen wird.

Inhalt aus Datei abrufen?

Gibt an, ob der Text der HTTP-Anforderung aus einer lokalen Datei auf dem Host, auf dem der Kontaktpunkt ausgeführt wird, abgerufen werden soll.

Werte

- Ausgewählt: Zeigt an, dass der Text der HTTP-Anforderung aus einer lokalen Datei auf dem Host, auf dem der Kontaktpunkt ausgeführt wird, abgerufen werden soll.
- Deaktiviert: Zeigt an, dass der Text der HTTP-Anforderung aus dem Feld "Inhalt" abgerufen werden soll.

Pfad der Inhaltsdatei

Gibt den Pfad zu der lokalen Datei auf dem Host ein, auf dem der Kontaktpunkt ausgeführt wird. Die lokale Datei enthält den Text der HTTP-Anforderung.

Inhalt

Gibt den Text der HTTP-Anforderung an.

HTTP-Proxy-Informationen

Beschreibungen der Eingabeparameter finden Sie unter HTTP-Proxy-Informationen.

Informationen zu HTTP-Header

Beschreibungen der Eingabeparameter finden Sie unter Informationen zu HTTP-Header (siehe Seite 448).

Informationen zu HTTP-Cookies

Beschreibungen der Eingabeparameter finden Sie unter Informationen zu HTTP-Cookies (siehe Seite 449).

Informationen zum Inhalt der HTTP-Antwort

Beschreibungen der Eingabeparameter finden Sie unter Informationen zum Inhalt der HTTP-Antwort (siehe Seite 449).

Informationen zur HTTP-Konfiguration

Beschreibungen der Eingabeparameter finden Sie unter Informationen zur HTTP-Konfiguration (siehe Seite 451).

Ausgabeparameter

HttpRemoteURL

HttpValidateSSLCert

HttpSvrAuth

HttpSvrNtlmAuth

HttpAuthUser

HttpAuthPwd

HttpAuthDomain

HttpReqIsChunked

HttpReqContentType

HttpReqContentCharset

HttpReqContentFromFile

HttpReqContentFilePath

HttpReqContent

HttpProxy

HttpProxyHost

HttpProxyPort

HttpProxyAuth

HttpProxyNtlmAuth

HttpProxyUser

HttpProxyPwd

HttpProxyDomain

HttpHeaderFieldsType

HttpHeaderFieldsValueMap

HttpHeaderFieldsVarValueMap

HttpCookieFieldsVarValueMap

HttpRespSaveToFile

HttpRespLocalFile

HttpRespLocalFileExists

HttpRespSaveToDataset

HttpRespDatasetVarLimit

HttpVersion

HttpConnectionTimeout

HttpSocketTimeout

HttpHandleRedirects

HttpMaxRedirects

HTTPRequestUrl

HTTPResponseStatusLine

HTTPResponseStatusCode

HTTPResponseReasonPhrase

HTTPResponseProtocolVersion

HTTPResponseContentType

HTTPResponseContentCharset

HTTPResponseContentLength

HTTPResponseContentEncoding

HTTPResponseContentFilePath

HTTPResponseContent

HTTPResponseHtmlContent

HTTPResponseHeaders

HTTPResponseContentIsChunked

HTTPRequestHeaders

HTTPRequestLine

HTTPCookiesStore

Weitere Informationen finden Sie unter HTTP-Operatoren: Allgemeine Ausgabeparameter (siehe Seite 453).

Ausgabe-Ports

Ausgabe-Ports für den Operator "HTTP POST" enthalten allgemeine Ausgabe-Ports für HTTP-Operatoren und einen zusätzlichen Fehlerfall.

Operator "HTTP TRACE"

Der Operator "HTTP TRACE" sendet ein "HTTP TRACE" in eine URL. Die Methode "TRACE" fordert an, dass der HTTP-Server die empfangene Anfrage zurücksendet. Dieser Prozess kann für Testzwecke nützlich sein und Änderungen der Anfrage identifizieren, die von Proxys vorgenommen wurden. Die Anfrage wird als Inhalt der Antwort zurückgesendet.

Verwenden Sie den Operator "HTTP OPTIONS", um zu überprüfen, ob "HTTP TRACE" aktiviert ist.

Eingabeparameter

In folgenden Abschnitten finden Sie Beschreibungen der Eingabeparameter für den Operator "HTTP OPTIONS":

Informationen zu HTTP-URL

Beschreibungen der Eingabeparameter finden Sie unter Informationen zu HTTP-URL (siehe Seite 440).

HTTP-Proxy-Informationen

Beschreibungen der Eingabeparameter finden Sie unter HTTP-Proxy-Informationen (siehe Seite 444).

Informationen zu HTTP-Header

Beschreibungen der Eingabeparameter finden Sie unter Informationen zu HTTP-Header (siehe Seite 448).

Informationen zu HTTP-Cookies

Beschreibungen der Eingabeparameter finden Sie unter Informationen zu HTTP-Cookies (siehe Seite 449).

Informationen zum Inhalt der HTTP-Antwort

Beschreibungen der Eingabeparameter finden Sie unter Informationen zum Inhalt der HTTP-Antwort (siehe Seite 449).

Informationen zur HTTP-Konfiguration

Beschreibungen der Eingabeparameter finden Sie unter Informationen zur HTTP-Konfiguration (siehe Seite 451).

Ausgabeparameter

HttpRemoteURL

HttpValidateSSLCert

HttpSvrAuth

HttpSvrNtlmAuth

HttpAuthUser

HttpAuthPwd

HttpAuthDomain

HttpProxy

HttpProxyHost

HttpProxyPort

HttpProxyAuth

HttpProxyNtlmAuth

HttpProxyUser

HttpProxyPwd

HttpProxyDomain

HttpHeaderFieldsType

HttpHeaderFieldsValueMap

HttpHeaderFieldsVarValueMap

HttpCookieFieldsVarValueMap

HttpRespSaveToFile

HttpRespLocalFile

HttpRespLocalFileExists

HttpRespSaveToDataset

HttpRespDatasetVarLimit

HttpVersion

HttpConnectionTimeout

HttpSocketTimeout

HttpHandleRedirects

HttpMaxRedirects

HTTPRequestUrl

HTTPResponseStatusLine

HTTPResponseStatusCode

HTTPResponseReasonPhrase

HTTPResponseProtocolVersion

HTTPResponseContentType

HTTPResponseContentCharset

HTTPResponseContentLength

HTTPResponseContentEncoding

HTTPResponseContentFilePath

HTTPResponseContent

HTTPResponseHtmlContent

HTTPResponseHeaders

HTTPResponseContentIsChunked

HTTPRequestHeaders

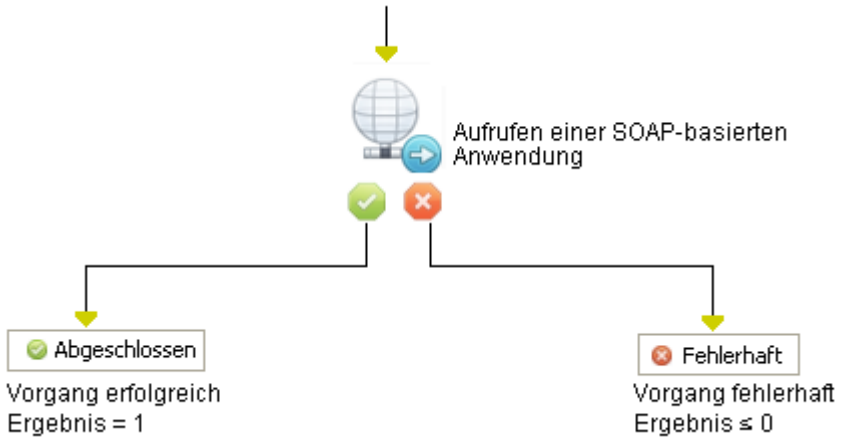
HTTPRequestLine

HTTPCookiesStore

Weitere Informationen finden Sie unter HTTP-Operatoren: Allgemeine Ausgabeparameter (siehe Seite 453).

Operator "SOAP-Methode aufrufen"

Der Operator "SOAP-Methode aufrufen" ruft SOAP-basierte Webservice-Methoden auf, entweder um eine Aktion auszulösen oder um Information abzurufen. EINE SOAP-Abfrage kann auch mit einer WSDL generiert werden.



Eingabeparameter

Eingabeparameter für den Operator "SOAP-Methode aufrufen" sind Folgende.

WSDL Explorer

WSDL-Assistent

Ein Assistent, mit dem Sie SOAP-Methoden auswählen können. Klicken Sie, um das Fenster "WSDL-Explorer" zu öffnen, in dem Sie Folgendes eingeben können:

WSDL-URL

- Geben Sie eine WSDL-URL ein, und klicken Sie auf Laden. Beispiele:

  `http://server:8080/itpam/soap?wsdl`

  `https://server:8443/itpam/soap?wsdl`

- Wählen Sie eine vor Kurzem eingegebene WSDL-URL aus der Drop-down-Liste aus, und klicken Sie auf Laden.

  Hinweis: Die Drop-down-Liste "WSDL-URL" enthält die letzten zehn Eingaben.

Der Assistent füllt die restlichen Felder auf. Nehmen Sie bei Bedarf Änderungen vor, indem Sie andere Eingaben aus den Drop-down-List auswählen und auf Übernehmen klicken.

WSDL-Services

WSDL-Ports

WSDL-Betriebsabläufe

- Verwenden Sie die Standardeingabe, die geladen wurde.
- Wählen Sie in der Drop-down-Liste eine Eingabe aus.

  Weitere Informationen zu den aufgelisteten Webservice-Methoden finden Sie im *Webservice-API-Referenzhandbuch*.

Antwortbereich, zum Beispiel:

```
<tns:AsyncSoapResponse xmlns:tns="http://www.ca.com/itpam">
<tns:MessageID>MessageID__</tns:MessageID>
<!--xsd:Any Type Here-->
</tns:AsyncSoapResponse>
```

Parameter von "SOAP-Aufrufdaten"

Dienst-URL

Definiert die URL für den SOAP-Service. Auf die URL wird normalerweise über HTTP oder HTTPS zugegriffen. Die URL ist normalerweise eine Einsprungstelle für eine oder mehrere Methoden.

Methodenname

Definiert die Methode oder Funktion, die auf dem SOAP-Server ausgeführt werden soll. Das Produkt übergibt diese Methode als MIME SOAPAction-Header an den SOAP-Dienst.

Authentifizierungstyp

Gibt an, wie der Aufruf auf dem SOAP-Server authentifiziert werden soll.

Dieser Parameter kann einen der folgenden Werte haben:

Keine Authentifizierung

Keine Authentifizierung des SOAP-Aufrufs.

SOAP-Basisauthentifizierung

Anwendername und Kennwort des berechtigten Anwenders für SOAP-Basisauthentifizierung verwenden.

HTTP-Authentifizierung

Anwendername und Kennwort des berechtigten Anwenders für HTTP-Authentifizierung verwenden. Wenn diese Funktion aktiviert ist und der autorisierte Anwender und das autorisierte Kennwort angegeben sind, verwendet das Produkt diese Anmeldeinformationen für HTTP-Basisauthentifizierung. WS-Sicherheit wird verwendet, wenn dies vom Anwender auf der WS-Sicherheit-Eingabeseite (siehe Seite 508) angegeben wurde.

NTLM-Authentifizierung

Verwenden Sie den autorisierten Anwendernamen, das autorisierte Kennwort und den autorisierten Domänennamen, die zum SOAP-Webservice-Server zugeordnet sind, um eine Verbindung zum SOAP-Server mit NTLM-Authentifizierung herzustellen.

SOAP Version (Produktversion)

Gibt die SOAP-Server-Version an, für die der Aufruf durchgeführt wird. Diese Option gibt einen Hinweis auf die zu Grunde liegende Logik, in der beim Erstellen der SOAP-Anfrage die entsprechende SOAP MessageFactory-Klasse verwendet wird.

Dieser Parameter kann einen der folgenden Werte haben:

- SOAP 1.1
- SOAP 1.2

Eingabequelle

Definiert die Quelle für die Eingabeanforderung des SOAP-Diensts.

Die Eingabeanforderung kann Makros und XPath-Zuweisungen enthalten, die dynamisch die SOAP-Abfrage zur Laufzeit ändern. Wenn erforderlich, ermöglichen es diese XPath-Zuweisungen, die SOAP-Abfrage mit zur Laufzeit abgerufenen Werten zu aktualisieren.

Die folgenden Methoden enthalten eine vollständige, richtig formatierte XML-Meldung, die einen SOAP-Umschlag enthalten kann:

Inline-Text

Mit dieser Option können Sie den Operator "SOAP-Methode aufrufen" verwenden, um die Eingabeaufforderung anzugeben. Um die formatierte SOAP-Eingabemeldung anzugeben, verwenden Sie den Parameter "Inline-Text".

Vorformatierte SOAP-Datei

Mit dieser Option können Sie die Eingabeaufforderung in einer vorformatierten SOAP-Datei angeben. Um den Pfad zu einer Datei anzugeben, die eine Meldung in einem gültigen XML-Dokument enthält, verwenden Sie den Parameter "Dateiname".

Ausdruck

Aktivieren Sie diese Option, wenn ein CA Process Automation-Ausdruck die SOAP-Anfrage enthält. Definieren Sie den CA Process Automation-Ausdruck im Parameter "Ausdruck".

Inline-Text

Führen Sie eine der folgenden Aktionen aus, wenn Sie den Parameter "Eingabequelle" auf "Inline-Text" festlegen:

- Um eine formatierte SOAP-Eingabemeldung zu verwenden, klicken Sie auf "(...)", um das Dialogfeld "Inline-Text" zu öffnen, und geben Sie die Meldung ein.
- Um eine SOAP-Meldung aus einer Textdatei einzulesen, klicken Sie auf "(...)", und wählen Sie die Datei aus einem lokalen Laufwerk oder Netzwerklaufwerk aus.

Dateiname

Definiert den vollqualifizierten Namen einer entsprechenden Datei. Um diesen Parameter zu verwenden, legen Sie "Eingabequelle" auf "Vorformatierte SOAP-Datei" fest. Das Produkt qualifiziert den Dateinamen in Relation zum Kontaktpunkt, der die Webservice-Operatoren ausführt.

Ausdruck

Definiert einen CA Process Automation-Ausdruck, aus dem die SOAP-Anfrage extrahiert werden soll.

Gespeicherte Aufrufdatei

Definiert den vollständigen Pfad der Datei, in der die letzte ausgehende SOAP-Anfrage abgespeichert werden soll. Verwenden Sie diese Datei, um zu prüfen, ob die Daten an die Webservice-Operatoren gesendet wurden. Wenn der Aufruf die Anfrage wegen falschen Werten ablehnt, verwenden Sie die Datei, um Fehler in der Anfrage zu beheben, bevor Sie sie erneut an die Webservice-Operatoren übergeben.

Dynamische Parameter

"Dynamische Parameter" stellen Aktualisierungswerte in einer SOAP-Abfrage bereit.

Parameterstil

Gibt die Methode an, mit der Werte in einer SOAP-Abfrage aktualisiert werden. Die Webservices-Operatoren unterstützen zur Laufzeit die folgenden Methoden:

XPath-Zuweisungen

Gibt an, dass XPath-Ausdrücke im Parameter "Parameterliste" die Werte in einer SOAP-Anfrage aktualisieren.

Makroerweiterung

Gibt an, dass Makros im Parameter "Parameterliste" die Werte in einer SOAP-Anfrage aktualisieren.

Parameterliste

Gibt die eindeutigen Dateneingaben in der SOAP-Anfrage an.

- Klicken Sie auf "Hinzufügen", um einen Parameter hinzuzufügen.
- Klicken Sie auf "Bearbeiten", um den derzeit ausgewählten Parameter zu ändern.
- Klicken Sie auf "Löschen", um den derzeit^ ausgewählten Parameter zu löschen.

Es hat sich bewährt, die Parameter in derselben Reihenfolge hinzuzufügen, in der sie in der WSDL-Datei für den SOAP-Aufruf angegeben sind.

Jeder Parameter enthält die folgenden Werte:

Makroname/XPath-Abfrage

Definiert den Namen des Makros bzw. der XPath-Abfrage. Wenn der Wert der Name eines Makros ist, ersetzt der Wert den Makronamen. Wenn der Wert eine XPath-Abfrage ist, aktualisiert der Wert den Knoten, den die Abfrage zurückgibt.

Wert

Gibt einen Laufzeitwert für den Parameter an.

Typ

Gibt einen der folgenden Datentypen für den Parameter an:

- Ganzzahlwert
- Zeichenfolgenwert
- ValueMap
- XML-Fragment

Parameter von "Aufrufergebnisse"

Die folgenden Parameter für "Aufrufergebnisse" legen fest, wie die Ergebnisse eines SOAP-Aufrufs gespeichert werden sollen:

Speicherort der Antwortdatei

Definiert den vollqualifizierten Pfad für die Datei, die die Antwort auf die SOAP-Anfrage wiederherstellt. Eine vorhandene Datei wird durch eine neue Antwort überschrieben.

Elemente erster Ebene des SOAP-Antworttexts zu einzelnen Datensatzvariablen extrahieren

Speichert das Element, das im Textkörper der SOAP-Antwort die erste Ebene einnimmt, in einer separaten Datensatzvariable, wenn die SOAP-Antwort 12 KB überschreitet.

SOAP-Antworttext zu Datensatzvariable extrahieren

Speichert der Textkörper der Antwort in einer Datensatzvariablen.

SOAP-Header zu Datensatzvariable extrahieren

Speichert den Header der Antwort in einer Datensatzvariablen.

SOAP-Textelemente erster Ebene zu einzelnen Datensatzvariablen extrahieren

Speichert die SOAP-Text-Elemente, die die erste Ebene einnehmen, in einer einzelnen Datensatzvariable.

XML-Namespaces aus Antwort entfernen

Entfernt die Namespaces von Antworten, sodass Anwender einfachere XPath-Ausdrücke angeben können, um den Wert eines bestimmten Elements zu suchen. Diese Option ist in allen SOAP-Operatoren verfügbar.

Zusätzliche extrahierte Daten (aus vollständiger Antwort)

Gibt XPath-Ausdrücke an, die Daten aus dem Textkörper der SOAP-Antwort extrahieren. Geben Sie für jeden angegebenen Ausdruck eine Datensatzvariable, in der die extrahierten Daten gespeichert werden sollen, und einen Datentyp an. Klicken Sie auf "Hinzufügen", um einen Ausdruck hinzuzufügen, auf "Bearbeiten", um einen ausgewählten Ausdruck zu ändern, und auf "Löschen", um einen ausgewählten Ausdruck zu löschen. Die Sortierung der Ausdrücke hat keine Bedeutung für CA Process Automation.

Jeder Parameter enthält die folgenden Werte:

XPath Ausdruck

Gibt den XPath-Ausdruck an.

Datensatzvariable

Gibt den Namen einer Operatordatensatzvariablen an, in der Werte gespeichert werden sollen, die anhand des ausgewählten XPath-Ausdrucks extrahiert wurden.

Typ

Gibt den Typ der Elemente an, die aus der Antwort extrahiert werden. Wählen Sie einen der folgenden gegenwärtig unterstützten Typen aus:

- Ganzzahl
- Zeichenfolge
- Array mit ganzen Zahlen
- Zeichenfolgen-Array
- XML-Fragment
- XML-Fragment-Array

MIME-Anhänge

Wenn der Inhalt, den Sie senden möchten, bereits in einer Datensatzvariable ist, verwenden Sie das Feld "Ausdruck".

Ausdruck?

Ist diese Option ausgewählt, muss ein Ausdruck den Anhang auflösen.

Ausdruck

Definiert den MIME-Ausdruck (Multipurpose Internet Mail Extensions), der den Anhang aus dem Text der SOAP-Antwort extrahiert. Geben Sie für jeden definierten Ausdruck eine Datensatzvariable, in der die extrahierten Daten gespeichert werden sollen, sowie einen Datentyp an. Die Sortierung der Ausdrücke hat keine Bedeutung für CA Process Automation. Klicken Sie auf "Hinzufügen", um einen Anhang hinzuzufügen, auf "Bearbeiten", um einen ausgewählten Anhang zu ändern, und auf "Löschen", um einen ausgewählten Anhang zu löschen.

Inhaltstyp

Definiert den Inhaltstyp des MIME-Anhangs (zum Beispiel "Text").

Inhalts-ID

Definiert die eindeutige ID für den MIME-Anhang.

Datei-URL

Definiert die URL des MIME-Anhangs.

WS-Sicherheit

Web Services-Sicherheit (WS) ermöglicht es CA Process Automation, sicheren SOAP-Meldungsaustausch mit einem Webservice durchzuführen, der zusätzliche Sicherheit benötigt.

WS-Sicherheit-Funktionen:

- Timestamps (Zeitstempel)
- UsernameTokens (Benutzernamentoken)
- Signatures (Signaturen)
- Verschlüsselung

Hinweis: Die WS-Sicherheit-Parameter können nur in den Operatoren festgelegt werden. Es sind keine Parameter mit Operatorkategorie für WS-Sicherheit verfügbar.

Nachdem sie definiert wurden, schließt der Tag <wsse:Security> des Header der SOAP-Anfrage die folgenden Parameter (oder Teile dieser Parameter) ein. Der Webservice führt dann Folgendes aus:

- Überprüft die Parameter zur Authentifizierung.
- Stellt sicher, dass die SOAP-Anfrage während der Übertragung vom Client zum Server nicht geändert wurde.

Allgemeine WS-Sicherheit-Parameter

Actor

Legt das Attribut "Actor" des Header <wsse:Security> der SOAP-Anfrage fest. Wenn Sie eines der folgenden Attribute festlegen, legt das Produkt das Attribut "Actor" fest:

- Verschlüsselung hinzufügen
- Signatur hinzufügen
- Zeitstempel hinzufügen
- Benutzernamentoken hinzufügen

Hinweis: Lassen Sie diesen Parameter leer, wenn kein Actor angegeben ist oder wenn Sie SOAP 1.2 verwenden.

mustUnderstand

Wenn Sie eines der folgenden Attribute festlegen, setzt dieser Parameter das Attribut "mustUnderstand" des Header <wsse:Security> auf "true":

- Verschlüsselung hinzufügen
- Signatur hinzufügen
- Zeitstempel hinzufügen
- Benutzernamentoken hinzufügen

Hinweis: Das Attribut "mustUnderstand" gibt an, ob die Eingabe des Header <wsse:Security> vom Empfänger obligatorisch oder optional bearbeitet werden muss.

Zeitstempel

Zeitstempel hinzufügen

Fügt dem Header <wsse:Security> einen Zeitstempel hinzu und aktiviert alle Zeitstempelparameter.

Zeitstempelparameter

Enthält die folgenden Werte:

Zeit bis Ablauf (Sek.)

Definiert die Zeit zwischen Erstellung und Ablauf der SOAP-Anfrage (in Sekunden). Wenn Sie dieses Feld leer lassen, wird standardmäßig 0 verwendet, und das Produkt richtet keine Ablaufzeit ein. Wenn der Zeitstempel abläuft, lehnt der Webservice die SOAP-Anfrage ab.

Zeitstempel-Präzision auf Millisekunden festlegen

Wenn diese Option aktiviert ist, legt das Produkt die Zeitstempel-Präzision auf Millisekunden fest.

Benutzernamentoken

Benutzernamentoken hinzufügen

Fügt dem Header <wsse:Security> ein Benutzernamentoken hinzu und aktiviert alle Benutzernamentoken-Parameter.

Benutzernamentoken-Parameter

Enthält die folgenden Werte:

Anwendername

Definiert den Wert des Benutzernamentoken.

Kennwort

Definiert das Kennwort, das mit dem Benutzernamentoken verbunden ist.

Kennworttyp

Gibt den Typ des Kennworts an.

Werte:

- **0:** Das Produkt gibt das Kennwort als Klartext an.
- **1:** Das Produkt gibt das Kennwort in Digest-Form an (das Kennwort wird verschlüsselt und nicht als Klartext geliefert).
- **2:** Das Produkt gibt kein Kennwort an.

Nonce hinzufügen?

Wenn diese Option aktiviert ist, fügt das Produkt ein Nonce-Element (wie z. B. einen Hash-Wert) zum Benutzernamentoken hinzu. Möglicherweise benötigt der Webservice dieses Element nicht.

Created hinzufügen?

Wenn diese Option aktiviert ist, fügt das Produkt ein Created-Element hinzu, das anzeigt, wann das Benutzernamentoken erstellt wurde. Möglicherweise benötigt der Webservice dieses Element nicht.

Hinweis: Wenn Sie den Kennworttyp auf "Digest" festlegen, fügt das Produkt die Elemente "Nonce" und "Created" automatisch zum Benutzernamenstoken hinzu.

Schlüsselspeicherparameter

Wenn Sie eine SOAP-Anfrage signieren oder verschlüsseln, benötigt das Produkt viele Schlüssel, um verschiedenen Verantwortungsbereichen gerecht zu werden. Das Produkt verwendet Schlüssel in WS-Sicherheit, um die SOAP-Anfrage zu signieren oder zu verschlüsseln und die Signatur oder Entschlüsselung der SOAP-Antwort bei Bedarf zu validieren.

CA Process Automation verwendet einen *Schlüsselspeicher* (ein Repository mit Sicherheitszertifikaten), um die Schlüssel zu verwalten, die in WS-Sicherheit verwendet werden. Mit dem Schlüsselspeicher können Schlüssel organisiert und konsolidiert werden, und es wird verhindert, dass andere Anwender auf die eindeutigen privaten Schlüssel zugreifen. Das Produkt erfordert, dass Sie einen Schlüsselspeicher erstellen (siehe Seite 511) oder einen vorhandenen Schlüsselspeicher verwenden.

WS-Sicherheit umfasst die folgenden Schlüsselspeicheroptionen:

- Signieren der SOAP-Anfrage (ganz oder teilweise). Der Schlüsselspeicher enthält die folgenden Elemente:
  - Den privaten Schlüssel, der zum Signieren der Abfrage verwendet werden soll.
  - Der zugeordnete öffentliche Schlüssel, mit dem der Empfänger die Signatur validiert. Das Produkt fügt der signierten SOAP-Anfrage eine Referenz zu diesem öffentlichen Schlüssel hinzu.
- Verschlüsseln der SOAP-Anfrage (ganz oder teilweise). Der Schlüsselspeicher enthält den öffentlichen Schlüssel, mit dem der symmetrische Schlüssel verschlüsselt wird. Das Produkt verwendet den symmetrischen Schlüssel, um die Anfrage zu verschlüsseln.
- Validieren der Signatur der SOAP-Antwort. Der Schlüsselspeicher enthält den öffentlichen Schlüssel, mit dem die Signatur bei Bedarf verschlüsselt wird.
- Entschlüsseln der SOAP-Antwort. Der Schlüsselspeicher enthält den privaten Schlüssel, mit dem die Antwort entschlüsselt wird.

Die folgenden Parameter definieren den Schlüsselspeicher, mit dem eine SOAP-Anfrage signiert bzw. verschlüsselt werden kann:

Schlüsselspeicherpfad für Signatur/Verschlüsselung

Gibt den Pfad zum Keystore an.

Schlüsselspeicherkennwort für Signatur/Verschlüsselung

Definiert das Klartextkennwort, mit dem auf den Schlüsselspeicher zugegriffen werden kann.

Hinweis: Dieser Parameter ist nicht das Kennwort für den Zugriff auf einen privaten Schlüssel im Schlüsselspeicher, sondern das Kennwort, um auf den Schlüsselspeicher selbst zuzugreifen.

Schlüsselspeichertyp für Signatur/Verschlüsselung

Gibt den Typ des Schlüsselspeichers an.

Werte:

- JKS
- PKCS12 (normalerweise mit einer .p12-Erweiterung)
- Leer (das Produkt verwendet den Standardwert)

Standard: JKS

Erstellen eines Schlüsselspeichers

Sie können ein Tool eines Drittanbieters verwenden, um Ihren Schlüsselspeicher zu erstellen oder um neue Zertifikate/Privatschlüssel in einen vorhandenen Schlüsselspeicher zu importieren. Ein Verwaltungstool für Schlüsselspeicher ist "Keytool", das mit Java JRE oder JDK geliefert wird. Einige Keytool-Befehle können hier gefunden werden:

- http://download.oracle.com/javase/1.5.0/docs/tooldocs/solaris/keytool.html
- http://download.oracle.com/javase/1.5.0/docs/tooldocs/windows/keytool.html

Sie können auch folgende Befehle ausführen:

```
keytool –help
```

Signatur

Signatur hinzufügen

Wenn diese Option aktiviert ist, signiert das Produkt die SOAP-Anfrage und fügt anschließend zum Header <wsse:Security> eine Signatur hinzu. Ein privater Schlüssel im Schlüsselspeicher signiert den Inhalt der SOAP-Anfrage. Das Produkt aktiviert auch alle Signaturparameter-Felder.

Signaturparameter

Die folgenden Parameter definieren die Signatur:

Alias des privaten Schlüssels

Definiert den Schlüsselalias im Schlüsselspeicher, den das Produkt für das Signieren verwendet.

Kennwort des privaten Schlüssels

Definiert das Schlüsselkennwort im Schlüsselspeicher.

Kanonisierungsalgorithmus

Definiert die Kanonisierungsmethode, die verwendet wird, um die Daten (den Text der SOAP-Anfrage oder jene Teile, die signiert werden müssen) zu serialisieren, bevor die Signatur angewendet wird. Lassen Sie dieses Feld leer, um ausschließlich den standardmäßigen XML-Kanonisierungsalgorithmus "xml-exc-c14n#" der Implementierung zu verwenden.

Signaturalgorithmus

Definiert den Algorithmus, mit dem die Signatur definiert werden soll. Lassen Sie dieses Feld leer, wenn Sie möchten, dass das Produkt versucht, einen Signaturalgorithmus zu entdecken und zu verwenden, der mit den Daten im Schlüssel übereinstimmt.

ID-Typ des öffentlichen Schlüssels

Gibt die Schlüssel-ID an, die die Identifizierungselemente des Zertifikats (des öffentlichen Schlüssels) in der Signatur bestimmt. Der Empfänger verwendet diesen Wert, um das Signaturzertifikat (den öffentlichen Schlüssel) zu identifizieren, das die Signatur der SOAP-Anfrage validiert.

Werte:

- **1 (Binäres Sicherheits-Token):** Das Produkt fügt <wsse:SecurityTokenReference> zum Element "Signature" hinzu. Das Element "Signature" verwendet ein URI-Fragment in einem <wsse:Reference>-Element, um auf das Signaturzertifikat (den öffentlichen Schlüssel) zu verweisen. Das URI-Fragment bezieht sich auf den öffentlichen Schlüssel der Signatur. Das Produkt schließt den öffentlichen Schlüssel der Signatur als binäre Daten in das Element <wsse:BinarySecurityToken> des Header <wsse:Security> ein.
- **2 (Name des Ausstellers und Seriennummer):** Das Produkt fügt <wsse:SecurityTokenReference> zum Element "Signature" hinzu. Das Element "Signature" verwendet das Element <ds:X509Data><ds:X509:IssuerSerial>, um auf das Signaturzertifikat (den öffentlichen Schlüssel) zu verweisen. Dieses Element identifiziert eindeutig ein Zertifikat durch den Namen des Ausstellers "X.509" und der Seriennummer.
- **3 (X509-Zertifikats-ID):** Das Produkt fügt <wsse:SecurityTokenReference> zum Element "Signature" hinzu. Das Element "Signature" verwendet das Element <wsse:KeyIdentifier ValueType="oasis-200401-wss-x509-token-profile-1.0#X509v3">, um auf das Signaturzertifikat (den öffentlichen Schlüssel) zu verweisen.
- **4 (Schlüssel-ID):** Das Produkt fügt <wsse:SecurityTokenReference> zum Element "Signature" hinzu. Das Element "Signature" verwendet das Element <wsse:KeyIdentifier ValueType="#oasis-200401-wss-x509-token-profile-1.0#X509SubjectKeyIdentifier">, um auf das Signaturzertifikat (den öffentlichen Schlüssel) zu verweisen.

Standard: 0. Der Operator verwendet die standardmäßige Schlüssel-ID (Name des Ausstellers und Seriennummer) aus der Implementierung.

Zu signierende Teile

Wählen Sie aus, welche Teile der SOAP-Anfrage signiert werden sollen. Klicken Sie auf "Parameter hinzufügen", um eine Sicherheits-ID oder eine Kombination aus Name und Namespace des zu signierenden Elements einzugeben.

Werte:

- **WSU-ID:** Definiert das Attribut "wsu:id" des Elements, das signiert werden soll. Sie können "wsu:id" als Attribut eines Elements in der SOAP-Abfrage hinzufügen und Ihren eigenen Wert angeben. Zum Beispiel:

  ```
  <token wsu:id="123"> </token>
  ```

  Die folgende Anweisung zeigt die Definition des WSU-Namespace:

```
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200
401-wss-wssecurity-utility-1.0.xsd"
```

- **Name:** Definiert den Namen des Elements, das signiert werden soll.
- **Namespace:** Definiert den Namespace-URI (nicht den lokalen Namen des Namespace) des Elements, das signiert werden soll. Zum Beispiel:

  ```
  "http://www.ca.com/itpam"
  ```

Hinweis: Lassen Sie diesen Parameter leer, um den Text der SOAP-Anfrage zu signieren. Wenn Sie die WSU-ID angeben, ignoriert das Produkt die Werte für Namen und Namespace.

Verschlüsselung

Verschlüsselung hinzufügen

Verschlüsselt die SOAP-Anfrage und fügt einen neuen verschlüsselten symmetrischen Schlüssel zum Header <wsse:Security> der SOAP-Anfrage hinzu. CA Process Automation verwendet einen symmetrischen Schlüssel, um den Inhalt der SOAP-Abfrage zu verschlüsseln. Das im Schlüsselspeicher angegebene Zertifikat (öffentlicher Schlüssel) verschlüsselt den symmetrischen Schlüssel selbst und schließt ihn in den Header <wsse:Security> ein. Wenn dieses Feld ausgewählt ist, werden alle Felder in den Verschlüsselungsparametern aktiviert.

Verschlüsselungsparameter

Die folgenden Parameter definieren die Verschlüsselung:

Alias des öffentlichen Schlüssels

Definiert den Alias des Zertifikats (des öffentlichen Schlüssels), mit dem der symmetrische Schlüssel im Schlüsselspeicher verschlüsselt werden soll.

Kanonisierungsalgorithmus

Definiert die Kanonisierungsmethode, die zum Serialisieren der Daten vor Anwendung der Verschlüsselung verwendet werden soll. Lassen Sie dieses Feld leer, um eine standardmäßige Serialisierung zu verwenden.

Algorithmus für symmetrische Verschlüsselung

Gibt den Typ des symmetrischen Algorithmus an, mit dem die Daten verschlüsselt werden sollen.

Werte:

- **Tripledes-cbc:** Verwendet Triple-DES. Diese Methode verwendet einen Schlüssel, der 8 Byte - 24 Bit lang ist.
- **aes128-cbc:** Verwendet AES mit einem 128-Bit-Schlüssel.
- **aes192-cbc:** Verwendet AES mit einem 192-Bit-Schlüssel.
- **aes256-cbc:** Verwendet AES mit einem 256-Bit-Schlüssel.

Standard: AES128

Hinweis: Wenn Sie diesen Parameter auf aes192-cbc oder aes256-cbc festlegen, kann der folgende Fehler auftreten:

```
Illegal key size or default parameters.
```

Wenn dieser Fehler auftritt, laden Sie die folgenden Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files von der Oracle-Website herunter:

- US_export_policy.jar
- local_policy.jar

Überschreiben Sie die vorhandenen JAR-Dateien mit demselben Namen auf C:*von_PAM_verwendeter_Pfad_zu_JRE*\lib\security mit den neuen Dateien.

Symmetrischen Schlüssel verschlüsseln?

Verschlüsselt den symmetrischen Schlüssel, mit dem die Daten verschlüsselt wurden. Das Produkt schließt den Schlüssel in den Header <wsse:security><xenc:EncryptedKey> ein.

Algorithmus für symmetrische Schlüssel-Verschlüsselung

Definiert den Algorithmus, mit dem der symmetrische Schlüssel verschlüsselt werden soll. Dieser Parameter ist nur anwendbar, wenn der Schlüssel verschlüsselt werden soll.

Standard: RSA15

ID-Typ des öffentlichen Schlüssels

Gibt die Schlüssel-ID an, die die Identifizierungselemente des Zertifikats (des öffentlichen Schlüssels) im Element <xenc:EncryptedKey> bestimmt. Der Empfänger verwendet den privaten Schlüssel, der diesem Zertifikat (öffentlicher Schlüssel) entspricht, um den symmetrischen Schlüssel zu entschlüsseln. Das Produkt verwendet den symmetrischen Schlüssel, um die SOAP-Anfrage zu entschlüsseln.

Werte:

- **1 (Binäres Sicherheits-Token):** Das Produkt fügt <wsse:SecurityTokenReference> zum Element <xenc:EncryptedKey> hinzu. Das Element <xenc:EncryptedKey> verwendet ein URI-Fragment in einem <wsse:Reference>-Element, um auf das Zertifikat (den öffentlichen Schlüssel) zu verweisen. Das URI-Fragment bezieht sich auf den öffentlichen Schlüssel. Das Produkt schließt den öffentlichen Schlüssel als binäre Daten in das Element <wsse:BinarySecurityToken> des Header <wsse:Security> ein.
- **2 (Name des Ausstellers und Seriennummer):** Das Produkt fügt <wsse:SecurityTokenReference> zum Element <xenc:EncryptedKey> hinzu. Das Element <xenc:EncryptedKey> verwendet das Element <ds:X509Data><ds:X509:IssuerSerial>, um auf das Zertifikat (den öffentlichen Schlüssel) zu verweisen. Dieses Element identifiziert eindeutig ein Zertifikat durch den Namen des Ausstellers "X.509" und der Seriennummer.
- **3 (X509-Zertifikats-ID):** Das Produkt fügt <wsse:SecurityTokenReference> zum Element <xenc:EncryptedKey> hinzu. Das Element <xenc:EncryptedKey> verwendet das Element <wsse:KeyIdentifier ValueType="oasis-200401-wss-x509-token-profile-1.0#X509v3">, um auf das Zertifikat (den öffentlichen Schlüssel) zu verweisen.
- **4 (Schlüssel-ID):** Das Produkt fügt <wsse:SecurityTokenReference> zum Element <xenc:EncryptedKey> hinzu. Das Element <xenc:EncryptedKey> verwendet das Element <wsse:KeyIdentifier ValueType="#oasis-200401-wss-x509-token-profile-1.0#X509SubjectKeyIdentifier">, um auf das Zertifikat (den öffentlichen Schlüssel) zu verweisen.
- **8 (Thumbprint SHA1-ID):** Das Produkt fügt <wsse:SecurityTokenReference> zum Element <xenc:EncryptedKey> hinzu. Das Element <xenc:EncryptedKey> verwendet das Element <wsse:KeyIdentifier ValueType="#oasis-wss-soap-message-security-1.1#ThumbprintSHA1">, um auf das Zertifikat (den öffentlichen Schlüssel) zu verweisen.

Standard: 0. Der Operator verwendet die standardmäßige Schlüssel-ID (Name des Ausstellers und Seriennummer) aus der Implementierung.

Zu verschlüsselnde Teile

Gibt an, welche Teile der SOAP-Anfrage verschlüsselt werden sollen. Klicken Sie auf "Parameter hinzufügen", um entweder eine Sicherheits-ID (WSU-ID) oder eine Kombination aus Name und Namespace des zu verschlüsselnden Elements einzugeben.

Werte:

- **WSU-ID:** Definiert das Attribut "wsu:id" des Elements, das verschlüsselt werden soll. Sie können "wsu:id" als Attribut eines Elements in der SOAP-Abfrage hinzufügen und Ihren eigenen Wert angeben. Zum Beispiel:

  ```
  <token wsu:id="123"> </token>
  ```

  Die folgende Anweisung zeigt die Definition des WSU-Namespace:

  ```
  xmlns:wsu=http://docs.oasis-open.org/wss/2004/01/oasis-2004
  01-wss-wssecurity-utility-1.0.xsd
  ```

- **Name:** Definiert den Namen des Elements, das verschlüsselt werden soll.
- **Namespace:** Definiert den Namespace-URI (nicht den lokalen Namen des Namespace) des Elements, das verschlüsselt werden soll. Zum Beispiel:

  ```
  http://www.ca.com/pam
  ```

- **Kodieren:** Wählen Sie "Inhalt" aus, um den Inhalt des Elements zu verschlüsseln, oder wählen Sie "Element" aus, um das gesamte Element zu verschlüsseln.

Hinweis: Lassen Sie dieses Feld leer, um den Textinhalt der SOAP-Abfrage zu verschlüsseln. Wenn Sie "WSU-ID" angeben, ignoriert das Produkt die Werte für Namen und Namespace.

Zuerst signieren?

Gibt an, ob die Signatur angewendet werden soll, bevor die Daten verschlüsselt werden. Dieser Parameter ist hilfreich, wenn das Produkt dieselben Daten verschlüsselt und in der SOAP-Anfrage signiert.

Signatur der SOAP-Antwort entschlüsseln und validieren

Gibt an, ob der Inhalt der SOAP-Antwort entschlüsselt und gegebenenfalls die Signatur bestätigt werden soll. Wenn Sie diese Option auswählen, aktiviert das Produkt den Parameter "Entschlüsselungskennwort des privaten Schlüssels".

Entschlüsselungskennwort des privaten Schlüssels

Definiert das Kennwort, mit dem auf den privaten Entschlüsselungsschlüssel im Schlüsselspeicher zugegriffen wird. Verwenden Sie dieses Kennwort, um auf den privaten Schlüssel zuzugreifen, mit dem eine verschlüsselte SOAP-Antwort entschlüsselt wird.

Verschlüsselungs-/Signaturprozess für die SOAP-Abfrage

Die Operatoren "SOAP-Methode aufrufen" und "Asynchrone SOAP-Methode aufrufen" signieren oder verschlüsseln den Text der SOAP-Anfrage zur Gänze oder teilweise.

Verschlüsselung

1. CA Process Automation verwendet den Kanonisierungsalgorithmus, um die Daten, die verschlüsselt werden sollen, zu serialisieren. Diese Daten sind entweder der gesamte Text der Anfrage oder jene Teile, die Sie in "Zu verschlüsselnde Teile" angeben.
2. CA Process Automation verwendet einen symmetrischen Schlüssel (abhängig vom Algorithmus für symmetrische Verschlüsselung), um die Daten zu verschlüsseln. Diese Daten sind entweder der gesamte Text der Anfrage oder jene Teile, die Sie in "Zu verschlüsselnde Teile" angeben.
3. CA Process Automation verwendet den Alias des öffentlichen Schlüssels, um den öffentlichen Schlüssel aus dem Schlüsselspeicher abzurufen. Der öffentliche Schlüssel verwendet den Algorithmus, den Sie in "Algorithmus für die Verschlüsselung des symmetrischen Schlüssels" angeben, um den symmetrischen Schlüssel zu verschlüsseln.
4. CA Process Automation fügt den verschlüsselten symmetrischen Schlüssel der SOAP-Abfrage im Element <xenc:EncryptedKey> hinzu.
5. Je nach dem Referenztyp, den Sie in "ID-Typ des öffentlichen Schlüssels" angegeben haben, fügt CA Process Automation der SOAP-Anfrage eine Referenz auf den öffentlichen Schlüssel hinzu, die verwendet wird, um den symmetrischen Schlüssel zu verschlüsseln.
6. Der Empfänger entschlüsselt den symmetrischen Schlüssel und verwendet anschließend den entschlüsselten symmetrischen Schlüssel, um die SOAP-Anfrage zu entschlüsseln.

Signatur

1. CA Process Automation verwendet "Alias des privaten Schlüssels" und "Kennwort des privaten Schlüssels", um den privaten Schlüssel aus dem Schlüsselspeicher abzurufen.
2. CA Process Automation verwendet den Kanonisierungsalgorithmus, um die zu verschlüsselnden Daten zu signieren. Diese Daten sind entweder der gesamte Text der Anfrage oder jene Teile, die Sie in "Zu signierende Teile" angeben.
3. CA Process Automation verwendet den privaten Schlüssel im Schlüsselspeicher, um den Inhalt der SOAP-Anfrage mithilfe des Signaturalgorithmus zu signieren, den der Anwender angibt. Das Produkt signiert entweder den gesamten Text der Anfrage oder jene Teile, die Sie in "Zu signierende Teile" angeben.

4. CA Process Automation fügt der SOAP-Anfrage eine Referenz zum Zertifikat bzw. zum öffentlichen Schlüssel, der zum privaten Schlüssel zugeordnet ist, hinzu. Der Anwender gibt den Typ der Referenz in "ID-Typ des öffentlichen Schlüssels" an.
5. Der Empfänger verwendet den öffentlichen Schlüssel, um die Signatur in der SOAP-Abfrage zu validieren.

Fehlerbehebung bei WS-Sicherheit

Die Operatoren "SOAP-Methode aufrufen" und "Asynchrone SOAP-Methode aufrufen" können fehlschlagen, wenn in den folgenden Fällen WS-Sicherheit angewendet wird.

Es kann kein Crypto für die Verwaltung von Schlüsselspeichern erstellt werden

Ein Fehler wird angezeigt, wenn der Operator keinen Crypto erstellen kann, um den Schlüsselspeicher zu verwalten:

```
SOAP invocation failed: Unable to build a Crypto for the keystore containing the Signature/Encryption keys…
```

Dieser Fehler kann unter folgenden Bedingungen auftreten:

- Ungültiges Schlüsselspeicherkennwort
- Ungültiger Schlüsselspeichertyp
- Der Schlüsselspeicherpfad verweist auf eine Datei, die kein Schlüsselspeicher ist.

Überwachen Sie die CA Process Automation-Protokolle im Fehler-Stack, um die Art des Problems zu erhalten. Zum Beispiel:

- Ungültiges Schlüsselspeicherkennwort: Schlüsselspeicher wurde geändert oder das Kennwort ist falsch.
- Ungültiger Schlüsselspeichertyp: java.security.KeyStoreException: *x* wurde nicht gefunden (wobei *x* der angegebene Typ ist)
- Schlüsselspeicherpfad verweist auf eine Datei, die kein Schlüsselspeicher ist: Ungültiges Schlüsselspeicherformat.

Signaturfehler

Bei der Signatur der SOAP-Abfrage können Probleme auftreten, wenn:

- Der Schlüsselspeicherpfad ist nicht auf dem Computer vorhanden, der den Kontaktpunkt enthält.
- Der Schlüsselspeicherpfad fehlt.
- Der Alias des privaten Schlüssels fehlt.
- Sie versuchen einen nicht existierenden Teil der SOAP-Abfrage zu signieren. Das Feld "Grund" enthält eine Meldung:

  `Element to encrypt/sign not found…`
- Der Alias des privaten Schlüssels ist im Schlüsselspeicher nicht vorhanden. Das Feld "Grund" enthält eine Meldung:

  `No certificates for user` *x* `were found for signature…`

  Wobei *x* der angegebene Alias des privaten Schlüssels ist.
- Das Kennwort des privaten Schlüssels ist erforderlich, aber es ist nicht angegeben oder das angegebene Kennwort ist falsch. Der Grund enthält eine Meldung:

  `Cannot recover key…`
- Sie geben einen ungültigen Kanonisierungsalgorithmus an. Das Feld "Grund" enthält eine Meldung:

  `Unknown canonicalizer. No handler installed for URI` *x*`…`

  Wobei *x* der Name des angegebenen Kanonisierungsalgorithmus ist.
- Sie geben einen ungültigen Signaturalgorithmus an. Das Feld "Grund" enthält eine Meldung:

  `The requested algorithm` *x* `does not exist...`

  Wobei *x* der Name des angegebenen Signaturalgorithmus ist.
- Sie geben einen ungültigen ID-Typ des öffentlichen Schlüssels an. Das Feld "Grund" enthält eine Meldung:

  `Unsupported key identification…`

Verschlüsselungsfehler

Bei der Verschlüsselung der SOAP-Abfrage können Probleme auftreten, wenn:

- Der Schlüsselspeicherpfad ist nicht auf dem Computer vorhanden, der den Kontaktpunkt enthält.
- Der Schlüsselspeicherpfad fehlt.
- Sie versuchen einen nicht existierenden Teil der SOAP-Abfrage zu verschlüsseln. Das Feld "Grund" enthält eine Meldung:

  `Element to encrypt/sign not found…`
- Sie versuchen den Algorithmus für symmetrische Verschlüsselung "aes192-cbc" oder "aes256-cbc" zu verwenden, ohne dabei ein Upgrade für die JAR-Dateien der "Unlimited Strength Jurisdiction Policy" durchzuführen. Das Feld "Grund" enthält eine Meldung:

  `Illegal key size or default parameters.`
- Der Alias des öffentlichen Schlüssels ist im Schlüsselspeicher nicht vorhanden. Das Feld "Grund" enthält eine Meldung:

  `No certificates for user` *x* `were found for encryption…`

  Wobei *x* der angegebene Alias des öffentlichen Schlüssels ist.
- Sie geben einen ungültigen Algorithmus für Verschlüsselung an. Das Feld "Grund" enthält eine Meldung:

  `SOAP invocation failed: Unable to encrypt the SOAP message.null.`

  Die WSS4J-Bibliothek löst in diesem Fall einen Fehler im Nullzeiger aus.
- Sie geben einen ungültigen Algorithmus für symmetrische Schlüssel-Verschlüsselung an. Das Feld "Grund" enthält eine Meldung:

  `unsupported key transport encryption algorithm:` *x*

  Wobei *x* der angegebene Algorithmus für symmetrische Schlüssel-Verschlüsselung ist.
- Sie geben einen ungültigen ID-Typ des öffentlichen Schlüssels an. Das Feld "Grund" enthält eine Meldung:

  `Unsupported key identification…`

Hinweis: Wenn Sie einen Teil der SOAP-Abfrage verschlüsseln und anschließend signieren, stellen Sie sicher, den Teil als Inhalt zu verschlüsseln. Dadurch wird sichergestellt, dass "wsu:id" (oder Name und Namespace) *nach* Verschlüsselung und *vor* der Signatur in der SOAP-Abfrage verbleiben. Anderenfalls wird dieser Teil beim Versuch der Signatur nicht gefunden.

Entschlüsselungsfehler

Ein ungültiges Kennwort, dass im Feld "Entschlüsselungskennwort des privaten Schlüssels" angegeben ist, führt dazu, dass der Operator fehlschlägt. Das Feld "Grund" enthält eine Meldung:

```
Unable to apply WSS security on incoming message (SOAP Response).
```

Die Signatur oder Entschlüsselung ist ungültig. Die geschachtelte Ausnahme ist: java.security.

```
UnrecoverableKeyException: Get Key failed:
```

Der angegebene letzte Block ist nicht richtig aufgefüllt.

Ausgabeparameter

Der Operator "SOAP-Methode aufrufen" schließt die folgenden Ausgabeparameter ein.

SOAP-Aufrufdaten

serviceURL

Gibt die Service-URL, die der SOAP-Aufruf verwendet, zurück.

methodName

Gibt den Namen der aufzurufenden Methode zurück.

userName

Gibt den Anwendernamen für HTTP-Basisauthentifizierung zurück.

password

Gibt das Kennwort für HTTP-Basisauthentifizierung zurück.

httpAuth

Zeigt je nach Ihrer Angabe für "HTTP-Basisauthentifizierung verwenden?" "wahr" oder "falsch" an.

soapVersion

Gibt die SOAP-Version zurück, die das Produkt für SOAP-Aufrufe verwendet; entweder SOAP 1.1 oder SOAP 1.2.

inputSource

Gibt einen der folgenden Werte für die Eingabequelle des SOAP-Aufrufs zurück:

- InlineText
- Vorformatierte SOAP-Datei
- Ausdruck

inlineText

Enthält die Inline-Textdaten, die für den SOAP-Aufruf verwendet werden. Diese Variable wird nur aufgefüllt, wenn Sie "Inline-Text" als die Eingabequelle auswählen.

Beispiel:

```
<checkServerStatus xmlns="http://www.ca.com/itpam">
CA AuthMinder
<token>token__</token>
<user>user__</user>
<password>password__</password>
</auth>
</checkServerStatus>
```

FileName

Enthält die Inline-Textdaten, die für den SOAP-Aufruf verwendet werden. Diese Variable wird nur aufgefüllt, wenn Sie "Vorformatierte SOAP-Datei" als die Eingabequelle auswählen.

Expression Value

Enthält die Inline-Textdaten, die für den SOAP-Aufruf verwendet werden. Diese Variable wird nur aufgefüllt, wenn Sie "Ausdruck" als die Eingabequelle auswählen.

SavedCallFileName

Gibt den Dateinamen im Feld "Gespeicherte Aufrufdatei" zurück. Diese Datei enthält den SOAP-Umschlag, den das Produkt tatsächlich für SOAP-Aufrufe verwendet.

Beispiel:

```
<?xml version="1.0" encoding="UTF-8"?><SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><SOAP-ENV
:Header/><SOAP-ENV:Body><checkServerStatus
xmlns="http://www.ca.com/itpam">
CA
AuthMinder<token>token__</token><user>user__</user><password>pa
ssword__</password></auth></checkServerStatus></SOAP-ENV:Body><
/SOAP-ENV:Envelope>
```

Dynamische Parameter

Paramsyle

Gibt den Parameterstil zurück, den Sie im Feld "Dynamischer Parameter" ausgewählt haben: "Xpath-Zuweisungen" oder "MacroExpansion".

DynamicParamsInfo

Ein Array von ValueMaps; zeigt die Anzahl von ValueMaps an, die es enthält.

DynamicParamsdata

Gibt Abfrage, Wert und Typ zurück.

Parameter von "Aufrufergebnisse"

responseFileName

Gibt den Dateinamen zurück, den Sie im Feld "Antwortdatei" angegeben haben. Diese Datei enthält die Antwort von einem SOAP-Aufruf.

xPathQuery

Gibt die XPath-Abfrage zurück, die für das Extrahieren der Daten aus den Aufrufergebnissen definiert ist.

datasetVar

Gibt die Variable zurück, die für die extrahierten Aufrufergebnisse erstellt wird.

Typ

Gibt die Variable zurück, die den Datentyp enthält, der vom Anwender für die Aufrufergebnisse definiert wurde.

isExtractToDataSet

Gibt "wahr" zurück, wenn Sie "SOAP-Antworttext zu Datensatzvariable extrahieren" aktivieren. Andernfalls wird "falsch" zurückgegeben.

isExtractHeadersToDataSet

Gibt "wahr" zurück, wenn Sie "SOAP-Header zu Datensatzvariable extrahieren" aktivieren. Andernfalls wird "falsch" zurückgegeben.

isExtractHeadersToIndividualDataSet

Gibt "wahr" zurück, wenn Sie "SOAP-Header-Elemente erster Ebene zu einzelnen Datensatzvariablen extrahieren" aktivieren. Andernfalls wird "falsch" zurückgegeben.

isExtractToIndividualDataSet

Gibt "wahr" zurück, wenn Sie "SOAP-Header-Elemente erster Ebene zu einzelnen Datensatzvariablen extrahieren" aktivieren. Andernfalls wird "falsch" zurückgegeben.

isStripXMLNamespaces

Gibt "wahr" zurück, wenn Sie "XML-Namespaces aus Antwort entfernen" aktivieren. Andernfalls wird "falsch" zurückgegeben.

Betriebsablauf-Ergebnisse

Die Betriebsablauf-Ergebnisse geben die Ausgabe der SOAP-Aufrufergebnisse je nach den Parametern "Aufrufergebnisse", die Sie ausgewählt haben, zurück.

SOAPResponseBody

Speichert den vollständigen SOAP-Antworttext.

SOAPResponseHeaderData

Speichert den Antwort-Header.

SoapResponseHeader

Gibt die isolierten Daten der SOAP-Header-Antwort zurück.

SOAPResponseData

Gibt die isolierten Daten der SOAP-Aufrufergebnisse zurück.

MIME-Anhänge

isResolvedByExpression

Gibt "wahr" zurück, wenn Sie die Option "Ausdruck?" für den MIME-Anhang aktivieren. Andernfalls wird "falsch" zurückgegeben. Durch die Auswahl der Option "Ausdruck?" geben Sie an, dass der MIME-Anhang als Ausdruck angegeben wird, und dass sich dieser Ausdruck auf den MIME-Anhang bezieht.

Expressionfield

Gibt den Ausdruck zurück, der sich auf den MIME-Anhang bezieht.

attachmentFields

Gibt ein ValueMap-Array zurück, das die Anzahl der Elemente enthält, die es umfasst.

Inhaltstyp

Gibt den Inhaltstyp des MIME-Anhangs zurück.

ContentID

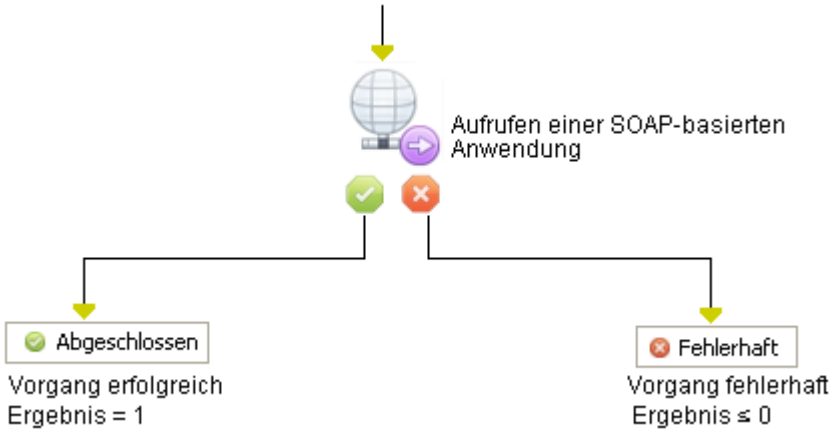
Gibt die Inhalts-ID zurück, die den MIME-Anhang eindeutig identifiziert.

FileURL

Gibt den Pfad des MIME-Anhangs zurück.

Operator "Asynchrone SOAP-Methode aufrufen"

Der Operator "Asynchrone SOAP-Methode aufrufen" unterstützt asynchrone SOAP-basierte Webservice-Methoden. Der Operator sendet in diesem Fall asynchron eine Nachricht an einen Empfänger und erwartet eine Antwort zu einem späteren Zeitpunkt. Der Absender kennzeichnet die Anfrage mit einem Bezeichner, der es ermöglicht, die Antwort mit der auslösenden Anfrage zu verknüpfen. Dieser Operator wird sehr ähnlich wie der Operator "SOAP-Methode aufrufen" verwendet und konfiguriert.



Eingabeparameter

Der Operator "Asynchrone SOAP-Methode aufrufen" stellt die folgenden Eingabeparameter bereit:

WSDL Explorer

WSDL-Assistent

Ein Assistent, mit dem Sie SOAP-Methoden auswählen können. Klicken Sie, um das Fenster "WSDL-Explorer" zu öffnen, in dem Sie Folgendes eingeben können:

WSDL-URL

- Geben Sie eine WSDL-URL ein, und klicken Sie auf Laden. Beispiele:

  `http://server:8080/itpam/soap?wsdl`

  `https://server:8443/itpam/soap?wsdl`

- Wählen Sie eine vor Kurzem eingegebene WSDL-URL aus der Drop-down-Liste aus, und klicken Sie auf Laden.

  Hinweis: Die Drop-down-Liste "WSDL-URL" enthält die letzten zehn Eingaben.

Der Assistent füllt die restlichen Felder auf. Nehmen Sie bei Bedarf Änderungen vor, indem Sie andere Eingaben aus den Drop-down-List auswählen und auf Übernehmen klicken.

WSDL-Services

WSDL-Ports

WSDL-Betriebsabläufe

- Verwenden Sie die Standardeingabe, die geladen wurde.
- Wählen Sie in der Drop-down-Liste eine Eingabe aus.

  Weitere Informationen zu den aufgelisteten Webservice-Methoden finden Sie im *Webservice-API-Referenzhandbuch*.

Antwortbereich, zum Beispiel:

```
<tns:AsyncSoapResponse xmlns:tns="http://www.ca.com/itpam">
<tns:MessageID>MessageID__</tns:MessageID>
<!--xsd:Any Type Here-->
</tns:AsyncSoapResponse>
```

Eigenschaften von "SOAP-Aufrufdaten"

Dienst-URL

Gibt die URL für den SOAP-Dienst an. Auf die URL wird normalerweise über HTTP oder HTTPS zugegriffen. Die URL ist normalerweise eine Einsprungstelle für eine oder mehrere Methoden.

Methodenname

Definiert die Methode oder Funktion, die gestartet werden soll. Das Produkt übergibt die Methode als MIME SOAPAction-Header an den SOAP-Dienst.

Authentifizierungstyp

Gibt einen der folgenden Authentifizierungstypen zum Aufrufen des SOAP-Servers an:

Keine Authentifizierung

Authentifizieren Sie sich nicht.

SOAP-Basisauthentifizierung

Anwendername und Kennwort des berechtigten Anwenders für SOAP-Basisauthentifizierung verwenden.

HTTP-Authentifizierung

Verwenden Sie den autorisierten Anwendernamen und das autorisierte Kennwort für HTTP-Basisauthentifizierung, wenn diese Funktion aktiviert ist. WS-Sicherheit wird verwendet, wenn dies vom Anwender auf der WS-Sicherheit-Eingabeseite (siehe Seite 508) angegeben wurde.

NTLM-Authentifizierung

Verwenden Sie den autorisierten Anwender, das autorisierte Kennwort und den Domänennamen, um die Verbindung zum SOAP-Server mit NTLM-Authentifizierung herzustellen.

Wichtig! Die angegebenen Werte für Anwendername, Kennwort und Domänenname beziehen sich auf den Server des SOAP-Webservice.

SOAP Version (Produktversion)

Gibt die SOAP-Server-Version an, die das Produkt für SOAP-Aufrufe verwendet; entweder SOAP 1.1 oder SOAP 1.2. Diese Option gibt einen Hinweis auf die zugrunde liegende Logik, die beim Erstellen der SOAP-Anfrage die geeignete SOAP MessageFactory-Klasse verwendet.

Eingabequelle

Gibt die Eingabequelle des SOAP-Service an. Alle folgenden Methoden enthalten eine vollständige, richtig formatierte XML-Meldung, die einen SOAP-Umschlag enthalten kann:

Inline-Text

Der Operator "SOAP-Client-Aufruf" definiert die Eingabeaufforderung. Wenn Sie diese Option wählen, verwenden Sie die Option "Inline-Text", um die formatierte SOAP-Eingabemeldung anzugeben.

Vorformatierte SOAP-Datei

Die Eingabeaufforderung wird durch eine vorformatierte SOAP-Datei definiert. Wenn Sie diese Option wählen, geben Sie den Pfad zur Datei im Feld "Dateiname" an. Stellen Sie sicher, dass die Datei eine Meldung in einem gültigen XML-Dokument enthält.

Ausdruck

Die SOAP-Anfrage wird durch einen CA Process Automation-Ausdruck definiert. Wenn Sie diese Option wählen, geben Sie den CA Process Automation-Ausdruck im Feld "Ausdruck" an.

Die Eingabeanforderung kann Makros und XPath-Zuweisungen enthalten, die dynamisch die SOAP-Abfrage zur Laufzeit ändern. Diese Makros und XPath-Zuordnungen ermöglichen es, dass das Produkt die SOAP-Anfrage bei Bedarf mit zur Laufzeit abgerufenen Werten aktualisiert.

Inline-Text

Klicken Sie auf "(...)", um das Dialogfeld "Inline-Text" zu öffnen, wenn Sie "Eingabequelle" auf "Inline-Text" festlegen. Geben Sie eine formatierte SOAP-Eingabemeldung ein. Um eine SOAP-Meldung aus einer Textdatei in einem lokalen Netzwerk oder Netzwerklaufwerk zu lesen, klicken Sie im Dialogfeld "Inline-Text" auf "Importieren aus Datei".

Dateiname

Definiert den vollqualifizierten Dateinamen für den Kontaktpunkt, auf dem die Webservice-Operatoren ausgeführt werden. Das Produkt verwendet diese Eigenschaft, als Sie "Eingabequelle" auf "Vorformatierte SOAP-Datei" festlegen.

Ausdruck

Definiert einen CA Process Automation-Ausdruck, aus dem die SOAP-Anfrage extrahiert werden soll.

Gespeicherte Aufrufdatei

Definiert den vollständigen Pfad zum SOAP-Umschlag, den das Produkt verwendet, um einen SOAP-Aufruf durchzuführen. Verwenden Sie diese Option, um zu prüfen, ob das Produkt die Daten an die Webservice-Operatoren gesendet hat. Wenn die SOAP-Anfrage abgelehnt wird, kann die gespeicherte Datei hilfreich sein, um die Fehler in der Anfrage zu beheben, bevor sie erneut an die Webservices-Operatoren gesendet wird.

Dynamische Parameter

Die dynamischen Parameter geben an, welche Werte in einer SOAP-Abfrage aktualisiert werden sollen.

Parameterstil

Gibt die Methode an, mit der die Parameterliste Werte in einer SOAP-Anfrage zur Laufzeit aktualisiert.

Makroerweiterung

Die Parameterliste verwendet Makros, um Werte in einer SOAP-Anfrage zu aktualisieren.

XPath-Zuweisungen

Die Parameterliste verwendet XPath-Ausdrücke, um Werte in einer SOAP-Anfrage zu aktualisieren.

Parameterliste

Gibt die Parameter an, die eindeutige Dateneinträge in der SOAP-Anfrage festlegen. Klicken Sie auf "Hinzufügen", um einen Parameter hinzuzufügen, auf "Bearbeiten", um den derzeit ausgewählten Parameter zu bearbeiten, und auf "Löschen", um den derzeit ausgewählten Parameter zu löschen. Es hat sich bewährt, die Parameter in derselben Reihenfolge hinzuzufügen, in der sie in der WSDL-Datei für den SOAP-Aufruf angegeben sind.

Um das Dialogfeld "Dynamische Parameter" zu öffnen, klicken Sie auf Hinzufügen" oder "Bearbeiten", und legen Sie die folgenden Parameterwerte fest:

Macro Name-X-Path

Definiert den Namen des Makros oder der XPath-Abfrage, das bzw. die verwendet wird, um Daten aus dem Text der SOAP-Antwort zu extrahieren.

- Wenn der Parameter einen Makronamen definiert, ersetzt der Wert den Makronamen.
- Wenn der Parameter eine XPath-Anfrage definiert, aktualisiert der Wert den Knoten, den die Abfrage zurückgibt.

Wert

Definiert einen Ausdruck, der einen Laufzeitwert für den Parameter zurückgibt.

Typ

Gibt einen der folgenden Datentypen für den Parameter an:

- Ganzzahlwert
- Zeichenfolgenwert
- XML-Fragment

Eigenschaften von "Aufrufergebnisse"

Mit den folgenden Parametern von "Aufrufergebnisse" wird festgelegt, wie die Ergebnisse eines SOAP-Aufrufs gespeichert werden sollen.

Speicherort der Antwortdatei

Gibt den vollständig qualifizierten Pfad für die Datei an, in der die Antwort auf die SOAP-Abfrage gespeichert wird. Die neue Antwort überschreibt vorhandene Antwortdateien.

Elemente erster Ebene des SOAP-Antworttexts zu einzelnen Datensatzvariablen extrahieren

Wenn die SOAP-Antwort 12 KB überschreitet, speichert dieser Parameter das Element erster Ebene im Text der SOAP-Antwort in einer separaten Datensatzvariable.

SOAP-Antworttext zu Datensatzvariable extrahieren

Speichert den Antworttext in einer Datensatzvariable.

SOAP-Header zu Datensatzvariable extrahieren

Speichert den Antwort-Header in einer Datensatzvariable.

SOAP-Header-Elemente erster Ebene zu einzelnen Datensatzvariablen extrahieren

Speichert den Antwort-Header in einzelnen Datensatzvariablen.

XML-Namespaces aus Antwort entfernen

Entfernt die XML-Namespaces aus der Antwort, sodass Sie die Lokal-name()-Funktion nicht verwenden müssen.

Zusätzliche extrahierte Daten (aus vollständiger Antwort)

Definiert XPath-Ausdrücke für die Extraktion von Daten aus dem SOAP-Antworttext. Geben Sie für jeden angegebenen Ausdruck einen Datentyp und eine Datensatzvariable zum Speichern der extrahierten Daten an. Klicken Sie auf "Hinzufügen", um einen Ausdruck hinzuzufügen, auf "Bearbeiten", um einen vorhandenen Ausdruck zu ändern, und auf "Löschen", um einen ausgewählten Ausdruck zu löschen. Die Reihenfolge der Ausdrücke hat in CA Process Automation keine Bedeutung.

Um die folgenden Parameter zu ändern, klicken Sie auf "Hinzufügen" oder "Bearbeiten":

XPath-Ausdruck

Definiert den XPath-Ausdruck.

Datensatzvariable

Definiert den Namen einer Operatordatensatzvariable, in der Werte gespeichert werden sollen, die anhand des ausgewählten XPath-Ausdrucks extrahiert wurden.

Typ

Gibt den Typ der Elemente an, die aus der Antwort extrahiert werden:

- Ganzzahl
- Zeichenfolge
- Array mit ganzen Zahlen
- Zeichenfolgen-Array
- XML-Fragment
- XML-Fragment-Array

MIME-Anhänge

Wenn der Inhalt, den Sie senden möchten, bereits in einer Datensatzvariable ist, verwenden Sie das Feld "Ausdruck".

Ausdruck?

Ist diese Option ausgewählt, muss ein Ausdruck den Anhang auflösen.

Ausdruck

Definiert den MIME-Ausdruck (Multipurpose Internet Mail Extensions), der den Anhang aus dem Text der SOAP-Antwort extrahiert. Geben Sie für jeden definierten Ausdruck eine Datensatzvariable, in der die extrahierten Daten gespeichert werden sollen, sowie einen Datentyp an. Die Sortierung der Ausdrücke hat keine Bedeutung für CA Process Automation. Klicken Sie auf "Hinzufügen", um einen Anhang hinzuzufügen, auf "Bearbeiten", um einen ausgewählten Anhang zu ändern, und auf "Löschen", um einen ausgewählten Anhang zu löschen.

Inhaltstyp

Definiert den Inhaltstyp des MIME-Anhangs (zum Beispiel "Text").

Inhalts-ID

Definiert die eindeutige ID für den MIME-Anhang.

Datei-URL

Definiert die URL des MIME-Anhangs.

WS-Sicherheit

Web Services-Sicherheit (WS) ermöglicht es CA Process Automation, sicheren SOAP-Meldungsaustausch mit einem Webservice durchzuführen, der zusätzliche Sicherheit benötigt.

WS-Sicherheit-Funktionen:

- Timestamps (Zeitstempel)
- UsernameTokens (Benutzernamentoken)
- Signatures (Signaturen)
- Verschlüsselung

Hinweis: Die WS-Sicherheit-Parameter können nur in den Operatoren festgelegt werden. Es sind keine Parameter mit Operatorkategorie für WS-Sicherheit verfügbar.

Nachdem sie definiert wurden, schließt der Tag <wsse:Security> des Header der SOAP-Anfrage die folgenden Parameter (oder Teile dieser Parameter) ein. Der Webservice führt dann Folgendes aus:

- Überprüft die Parameter zur Authentifizierung.
- Stellt sicher, dass die SOAP-Anfrage während der Übertragung vom Client zum Server nicht geändert wurde.

Allgemeine WS-Sicherheit-Parameter

Actor

Legt das Attribut "Actor" des Header <wsse:Security> der SOAP-Anfrage fest. Wenn Sie eines der folgenden Attribute festlegen, legt das Produkt das Attribut "Actor" fest:

- Verschlüsselung hinzufügen
- Signatur hinzufügen
- Zeitstempel hinzufügen
- Benutzernamentoken hinzufügen

Hinweis: Lassen Sie diesen Parameter leer, wenn kein Actor angegeben ist oder wenn Sie SOAP 1.2 verwenden.

mustUnderstand

Wenn Sie eines der folgenden Attribute festlegen, setzt dieser Parameter das Attribut "mustUnderstand" des Header <wsse:Security> auf "true":

- Verschlüsselung hinzufügen
- Signatur hinzufügen
- Zeitstempel hinzufügen
- Benutzernamentoken hinzufügen

Hinweis: Das Attribut "mustUnderstand" gibt an, ob die Eingabe des Header <wsse:Security> vom Empfänger obligatorisch oder optional bearbeitet werden muss.

Zeitstempel

Zeitstempel hinzufügen

Fügt dem Header <wsse:Security> einen Zeitstempel hinzu und aktiviert alle Zeitstempelparameter.

Zeitstempelparameter

Enthält die folgenden Werte:

Zeit bis Ablauf (Sek.)

Definiert die Zeit zwischen Erstellung und Ablauf der SOAP-Anfrage (in Sekunden). Wenn Sie dieses Feld leer lassen, wird standardmäßig 0 verwendet, und das Produkt richtet keine Ablaufzeit ein. Wenn der Zeitstempel abläuft, lehnt der Webservice die SOAP-Anfrage ab.

Zeitstempel-Präzision auf Millisekunden festlegen

Wenn diese Option aktiviert ist, legt das Produkt die Zeitstempel-Präzision auf Millisekunden fest.

Benutzernamentoken

Benutzernamentoken hinzufügen

Fügt dem Header <wsse:Security> ein Benutzernamentoken hinzu und aktiviert alle Benutzernamentoken-Parameter.

Benutzernamentoken-Parameter

Enthält die folgenden Werte:

Anwendername

Definiert den Wert des Benutzernamentoken.

Kennwort

Definiert das Kennwort, das mit dem Benutzernamentoken verbunden ist.

Kennworttyp

Gibt den Typ des Kennworts an.

Werte:

- **0:** Das Produkt gibt das Kennwort als Klartext an.
- **1:** Das Produkt gibt das Kennwort in Digest-Form an (das Kennwort wird verschlüsselt und nicht als Klartext geliefert).
- **2:** Das Produkt gibt kein Kennwort an.

Nonce hinzufügen?

Wenn diese Option aktiviert ist, fügt das Produkt ein Nonce-Element (wie z. B. einen Hash-Wert) zum Benutzernamentoken hinzu. Möglicherweise benötigt der Webservice dieses Element nicht.

Created hinzufügen?

Wenn diese Option aktiviert ist, fügt das Produkt ein Created-Element hinzu, das anzeigt, wann das Benutzernamentoken erstellt wurde. Möglicherweise benötigt der Webservice dieses Element nicht.

Hinweis: Wenn Sie den Kennworttyp auf "Digest" festlegen, fügt das Produkt die Elemente "Nonce" und "Created" automatisch zum Benutzernamenstoken hinzu.

Schlüsselspeicherparameter

Wenn Sie eine SOAP-Anfrage signieren oder verschlüsseln, benötigt das Produkt viele Schlüssel, um verschiedenen Verantwortungsbereichen gerecht zu werden. Das Produkt verwendet Schlüssel in WS-Sicherheit, um die SOAP-Anfrage zu signieren oder zu verschlüsseln und die Signatur oder Entschlüsselung der SOAP-Antwort bei Bedarf zu validieren.

CA Process Automation verwendet einen *Schlüsselspeicher* (ein Repository mit Sicherheitszertifikaten), um die Schlüssel zu verwalten, die in WS-Sicherheit verwendet werden. Mit dem Schlüsselspeicher können Schlüssel organisiert und konsolidiert werden, und es wird verhindert, dass andere Anwender auf die eindeutigen privaten Schlüssel zugreifen. Das Produkt erfordert, dass Sie einen Schlüsselspeicher erstellen (siehe Seite 511) oder einen vorhandenen Schlüsselspeicher verwenden.

WS-Sicherheit umfasst die folgenden Schlüsselspeicheroptionen:

- Signieren der SOAP-Anfrage (ganz oder teilweise). Der Schlüsselspeicher enthält die folgenden Elemente:
  - Den privaten Schlüssel, der zum Signieren der Abfrage verwendet werden soll.
  - Der zugeordnete öffentliche Schlüssel, mit dem der Empfänger die Signatur validiert. Das Produkt fügt der signierten SOAP-Anfrage eine Referenz zu diesem öffentlichen Schlüssel hinzu.
- Verschlüsseln der SOAP-Anfrage (ganz oder teilweise). Der Schlüsselspeicher enthält den öffentlichen Schlüssel, mit dem der symmetrische Schlüssel verschlüsselt wird. Das Produkt verwendet den symmetrischen Schlüssel, um die Anfrage zu verschlüsseln.
- Validieren der Signatur der SOAP-Antwort. Der Schlüsselspeicher enthält den öffentlichen Schlüssel, mit dem die Signatur bei Bedarf verschlüsselt wird.
- Entschlüsseln der SOAP-Antwort. Der Schlüsselspeicher enthält den privaten Schlüssel, mit dem die Antwort entschlüsselt wird.

Die folgenden Parameter definieren den Schlüsselspeicher, mit dem eine SOAP-Anfrage signiert bzw. verschlüsselt werden kann:

Schlüsselspeicherpfad für Signatur/Verschlüsselung

Gibt den Pfad zum Keystore an.

Schlüsselspeicherkennwort für Signatur/Verschlüsselung

Definiert das Klartextkennwort, mit dem auf den Schlüsselspeicher zugegriffen werden kann.

Hinweis: Dieser Parameter ist nicht das Kennwort für den Zugriff auf einen privaten Schlüssel im Schlüsselspeicher, sondern das Kennwort, um auf den Schlüsselspeicher selbst zuzugreifen.

Schlüsselspeichertyp für Signatur/Verschlüsselung

Gibt den Typ des Schlüsselspeichers an.

Werte:

- JKS
- PKCS12 (normalerweise mit einer .p12-Erweiterung)
- Leer (das Produkt verwendet den Standardwert)

Standard: JKS

Erstellen eines Schlüsselspeichers

Sie können ein Tool eines Drittanbieters verwenden, um Ihren Schlüsselspeicher zu erstellen oder um neue Zertifikate/Privatschlüssel in einen vorhandenen Schlüsselspeicher zu importieren. Ein Verwaltungstool für Schlüsselspeicher ist "Keytool", das mit Java JRE oder JDK geliefert wird. Einige Keytool-Befehle können hier gefunden werden:

- http://download.oracle.com/javase/1.5.0/docs/tooldocs/solaris/keytool.html
- http://download.oracle.com/javase/1.5.0/docs/tooldocs/windows/keytool.html

Sie können auch folgende Befehle ausführen:

```
keytool –help
```

Signatur

Signatur hinzufügen

Wenn diese Option aktiviert ist, signiert das Produkt die SOAP-Anfrage und fügt anschließend zum Header <wsse:Security> eine Signatur hinzu. Ein privater Schlüssel im Schlüsselspeicher signiert den Inhalt der SOAP-Anfrage. Das Produkt aktiviert auch alle Signaturparameter-Felder.

Signaturparameter

Die folgenden Parameter definieren die Signatur:

Alias des privaten Schlüssels

Definiert den Schlüsselalias im Schlüsselspeicher, den das Produkt für das Signieren verwendet.

Kennwort des privaten Schlüssels

Definiert das Schlüsselkennwort im Schlüsselspeicher.

Kanonisierungsalgorithmus

Definiert die Kanonisierungsmethode, die verwendet wird, um die Daten (den Text der SOAP-Anfrage oder jene Teile, die signiert werden müssen) zu serialisieren, bevor die Signatur angewendet wird. Lassen Sie dieses Feld leer, um ausschließlich den standardmäßigen XML-Kanonisierungsalgorithmus "xml-exc-c14n#" der Implementierung zu verwenden.

Signaturalgorithmus

Definiert den Algorithmus, mit dem die Signatur definiert werden soll. Lassen Sie dieses Feld leer, wenn Sie möchten, dass das Produkt versucht, einen Signaturalgorithmus zu entdecken und zu verwenden, der mit den Daten im Schlüssel übereinstimmt.

ID-Typ des öffentlichen Schlüssels

Gibt die Schlüssel-ID an, die die Identifizierungselemente des Zertifikats (des öffentlichen Schlüssels) in der Signatur bestimmt. Der Empfänger verwendet diesen Wert, um das Signaturzertifikat (den öffentlichen Schlüssel) zu identifizieren, das die Signatur der SOAP-Anfrage validiert.

Werte:

- **1 (Binäres Sicherheits-Token):** Das Produkt fügt <wsse:SecurityTokenReference> zum Element "Signature" hinzu. Das Element "Signature" verwendet ein URI-Fragment in einem <wsse:Reference>-Element, um auf das Signaturzertifikat (den öffentlichen Schlüssel) zu verweisen. Das URI-Fragment bezieht sich auf den öffentlichen Schlüssel der Signatur. Das Produkt schließt den öffentlichen Schlüssel der Signatur als binäre Daten in das Element <wsse:BinarySecurityToken> des Header <wsse:Security> ein.
- **2 (Name des Ausstellers und Seriennummer):** Das Produkt fügt <wsse:SecurityTokenReference> zum Element "Signature" hinzu. Das Element "Signature" verwendet das Element <ds:X509Data><ds:X509:IssuerSerial>, um auf das Signaturzertifikat (den öffentlichen Schlüssel) zu verweisen. Dieses Element identifiziert eindeutig ein Zertifikat durch den Namen des Ausstellers "X.509" und der Seriennummer.
- **3 (X509-Zertifikats-ID):** Das Produkt fügt <wsse:SecurityTokenReference> zum Element "Signature" hinzu. Das Element "Signature" verwendet das Element <wsse:KeyIdentifier ValueType="oasis-200401-wss-x509-token-profile-1.0#X509v3">, um auf das Signaturzertifikat (den öffentlichen Schlüssel) zu verweisen.
- **4 (Schlüssel-ID):** Das Produkt fügt <wsse:SecurityTokenReference> zum Element "Signature" hinzu. Das Element "Signature" verwendet das Element <wsse:KeyIdentifier ValueType="#oasis-200401-wss-x509-token-profile-1.0#X509SubjectKeyIdentifier">, um auf das Signaturzertifikat (den öffentlichen Schlüssel) zu verweisen.

Standard: 0. Der Operator verwendet die standardmäßige Schlüssel-ID (Name des Ausstellers und Seriennummer) aus der Implementierung.

Zu signierende Teile

Wählen Sie aus, welche Teile der SOAP-Anfrage signiert werden sollen. Klicken Sie auf "Parameter hinzufügen", um eine Sicherheits-ID oder eine Kombination aus Name und Namespace des zu signierenden Elements einzugeben.

Werte:

- **WSU-ID:** Definiert das Attribut "wsu:id" des Elements, das signiert werden soll. Sie können "wsu:id" als Attribut eines Elements in der SOAP-Abfrage hinzufügen und Ihren eigenen Wert angeben. Zum Beispiel:

  ```
  <token wsu:id="123"> </token>
  ```

  Die folgende Anweisung zeigt die Definition des WSU-Namespace:

```
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200
401-wss-wssecurity-utility-1.0.xsd"
```

- **Name:** Definiert den Namen des Elements, das signiert werden soll.
- **Namespace:** Definiert den Namespace-URI (nicht den lokalen Namen des Namespace) des Elements, das signiert werden soll. Zum Beispiel:

  ```
  "http://www.ca.com/itpam"
  ```

Hinweis: Lassen Sie diesen Parameter leer, um den Text der SOAP-Anfrage zu signieren. Wenn Sie die WSU-ID angeben, ignoriert das Produkt die Werte für Namen und Namespace.

Verschlüsselung

Verschlüsselung hinzufügen

Verschlüsselt die SOAP-Anfrage und fügt einen neuen verschlüsselten symmetrischen Schlüssel zum Header <wsse:Security> der SOAP-Anfrage hinzu. CA Process Automation verwendet einen symmetrischen Schlüssel, um den Inhalt der SOAP-Abfrage zu verschlüsseln. Das im Schlüsselspeicher angegebene Zertifikat (öffentlicher Schlüssel) verschlüsselt den symmetrischen Schlüssel selbst und schließt ihn in den Header <wsse:Security> ein. Wenn dieses Feld ausgewählt ist, werden alle Felder in den Verschlüsselungsparametern aktiviert.

Verschlüsselungsparameter

Die folgenden Parameter definieren die Verschlüsselung:

Alias des öffentlichen Schlüssels

Definiert den Alias des Zertifikats (des öffentlichen Schlüssels), mit dem der symmetrische Schlüssel im Schlüsselspeicher verschlüsselt werden soll.

Kanonisierungsalgorithmus

Definiert die Kanonisierungsmethode, die zum Serialisieren der Daten vor Anwendung der Verschlüsselung verwendet werden soll. Lassen Sie dieses Feld leer, um eine standardmäßige Serialisierung zu verwenden.

Algorithmus für symmetrische Verschlüsselung

Gibt den Typ des symmetrischen Algorithmus an, mit dem die Daten verschlüsselt werden sollen.

Werte:

- **Tripledes-cbc:** Verwendet Triple-DES. Diese Methode verwendet einen Schlüssel, der 8 Byte - 24 Bit lang ist.
- **aes128-cbc:** Verwendet AES mit einem 128-Bit-Schlüssel.
- **aes192-cbc:** Verwendet AES mit einem 192-Bit-Schlüssel.
- **aes256-cbc:** Verwendet AES mit einem 256-Bit-Schlüssel.

Standard: AES128

Hinweis: Wenn Sie diesen Parameter auf aes192-cbc oder aes256-cbc festlegen, kann der folgende Fehler auftreten:

```
Illegal key size or default parameters.
```

Wenn dieser Fehler auftritt, laden Sie die folgenden Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files von der Oracle-Website herunter:

- US_export_policy.jar
- local_policy.jar

Überschreiben Sie die vorhandenen JAR-Dateien mit demselben Namen auf C:*von_PAM_verwendeter_Pfad_zu_JRE*\lib\security mit den neuen Dateien.

Symmetrischen Schlüssel verschlüsseln?

Verschlüsselt den symmetrischen Schlüssel, mit dem die Daten verschlüsselt wurden. Das Produkt schließt den Schlüssel in den Header <wsse:security><xenc:EncryptedKey> ein.

Algorithmus für symmetrische Schlüssel-Verschlüsselung

Definiert den Algorithmus, mit dem der symmetrische Schlüssel verschlüsselt werden soll. Dieser Parameter ist nur anwendbar, wenn der Schlüssel verschlüsselt werden soll.

Standard: RSA15

ID-Typ des öffentlichen Schlüssels

Gibt die Schlüssel-ID an, die die Identifizierungselemente des Zertifikats (des öffentlichen Schlüssels) im Element <xenc:EncryptedKey> bestimmt. Der Empfänger verwendet den privaten Schlüssel, der diesem Zertifikat (öffentlicher Schlüssel) entspricht, um den symmetrischen Schlüssel zu entschlüsseln. Das Produkt verwendet den symmetrischen Schlüssel, um die SOAP-Anfrage zu entschlüsseln.

Werte:

- **1 (Binäres Sicherheits-Token):** Das Produkt fügt <wsse:SecurityTokenReference> zum Element <xenc:EncryptedKey> hinzu. Das Element <xenc:EncryptedKey> verwendet ein URI-Fragment in einem <wsse:Reference>-Element, um auf das Zertifikat (den öffentlichen Schlüssel) zu verweisen. Das URI-Fragment bezieht sich auf den öffentlichen Schlüssel. Das Produkt schließt den öffentlichen Schlüssel als binäre Daten in das Element <wsse:BinarySecurityToken> des Header <wsse:Security> ein.
- **2 (Name des Ausstellers und Seriennummer):** Das Produkt fügt <wsse:SecurityTokenReference> zum Element <xenc:EncryptedKey> hinzu. Das Element <xenc:EncryptedKey> verwendet das Element <ds:X509Data><ds:X509:IssuerSerial>, um auf das Zertifikat (den öffentlichen Schlüssel) zu verweisen. Dieses Element identifiziert eindeutig ein Zertifikat durch den Namen des Ausstellers "X.509" und der Seriennummer.
- **3 (X509-Zertifikats-ID):** Das Produkt fügt <wsse:SecurityTokenReference> zum Element <xenc:EncryptedKey> hinzu. Das Element <xenc:EncryptedKey> verwendet das Element <wsse:KeyIdentifier ValueType="oasis-200401-wss-x509-token-profile-1.0#X509v3">, um auf das Zertifikat (den öffentlichen Schlüssel) zu verweisen.
- **4 (Schlüssel-ID):** Das Produkt fügt <wsse:SecurityTokenReference> zum Element <xenc:EncryptedKey> hinzu. Das Element <xenc:EncryptedKey> verwendet das Element <wsse:KeyIdentifier ValueType="#oasis-200401-wss-x509-token-profile-1.0#X509SubjectKeyIdentifier">, um auf das Zertifikat (den öffentlichen Schlüssel) zu verweisen.
- **8 (Thumbprint SHA1-ID):** Das Produkt fügt <wsse:SecurityTokenReference> zum Element <xenc:EncryptedKey> hinzu. Das Element <xenc:EncryptedKey> verwendet das Element <wsse:KeyIdentifier ValueType="#oasis-wss-soap-message-security-1.1#ThumbprintSHA1">, um auf das Zertifikat (den öffentlichen Schlüssel) zu verweisen.

Standard: 0. Der Operator verwendet die standardmäßige Schlüssel-ID (Name des Ausstellers und Seriennummer) aus der Implementierung.

Zu verschlüsselnde Teile

Gibt an, welche Teile der SOAP-Anfrage verschlüsselt werden sollen. Klicken Sie auf "Parameter hinzufügen", um entweder eine Sicherheits-ID (WSU-ID) oder eine Kombination aus Name und Namespace des zu verschlüsselnden Elements einzugeben.

Werte:

- **WSU-ID:** Definiert das Attribut "wsu:id" des Elements, das verschlüsselt werden soll. Sie können "wsu:id" als Attribut eines Elements in der SOAP-Abfrage hinzufügen und Ihren eigenen Wert angeben. Zum Beispiel:

  ```
  <token wsu:id="123"> </token>
  ```

  Die folgende Anweisung zeigt die Definition des WSU-Namespace:

  ```
  xmlns:wsu=http://docs.oasis-open.org/wss/2004/01/oasis-2004
  01-wss-wssecurity-utility-1.0.xsd
  ```

- **Name:** Definiert den Namen des Elements, das verschlüsselt werden soll.
- **Namespace:** Definiert den Namespace-URI (nicht den lokalen Namen des Namespace) des Elements, das verschlüsselt werden soll. Zum Beispiel:

  ```
  http://www.ca.com/pam
  ```

- **Kodieren:** Wählen Sie "Inhalt" aus, um den Inhalt des Elements zu verschlüsseln, oder wählen Sie "Element" aus, um das gesamte Element zu verschlüsseln.

Hinweis: Lassen Sie dieses Feld leer, um den Textinhalt der SOAP-Abfrage zu verschlüsseln. Wenn Sie "WSU-ID" angeben, ignoriert das Produkt die Werte für Namen und Namespace.

Zuerst signieren?

Gibt an, ob die Signatur angewendet werden soll, bevor die Daten verschlüsselt werden. Dieser Parameter ist hilfreich, wenn das Produkt dieselben Daten verschlüsselt und in der SOAP-Anfrage signiert.

Signatur der SOAP-Antwort entschlüsseln und validieren

Gibt an, ob der Inhalt der SOAP-Antwort entschlüsselt und gegebenenfalls die Signatur bestätigt werden soll. Wenn Sie diese Option auswählen, aktiviert das Produkt den Parameter "Entschlüsselungskennwort des privaten Schlüssels".

Entschlüsselungskennwort des privaten Schlüssels

Definiert das Kennwort, mit dem auf den privaten Entschlüsselungsschlüssel im Schlüsselspeicher zugegriffen wird. Verwenden Sie dieses Kennwort, um auf den privaten Schlüssel zuzugreifen, mit dem eine verschlüsselte SOAP-Antwort entschlüsselt wird.

Verschlüsselungs-/Signaturprozess für die SOAP-Abfrage

Die Operatoren "SOAP-Methode aufrufen" und "Asynchrone SOAP-Methode aufrufen" signieren oder verschlüsseln den Text der SOAP-Anfrage zur Gänze oder teilweise.

Verschlüsselung

1. CA Process Automation verwendet den Kanonisierungsalgorithmus, um die Daten, die verschlüsselt werden sollen, zu serialisieren. Diese Daten sind entweder der gesamte Text der Anfrage oder jene Teile, die Sie in "Zu verschlüsselnde Teile" angeben.
2. CA Process Automation verwendet einen symmetrischen Schlüssel (abhängig vom Algorithmus für symmetrische Verschlüsselung), um die Daten zu verschlüsseln. Diese Daten sind entweder der gesamte Text der Anfrage oder jene Teile, die Sie in "Zu verschlüsselnde Teile" angeben.
3. CA Process Automation verwendet den Alias des öffentlichen Schlüssels, um den öffentlichen Schlüssel aus dem Schlüsselspeicher abzurufen. Der öffentliche Schlüssel verwendet den Algorithmus, den Sie in "Algorithmus für die Verschlüsselung des symmetrischen Schlüssels" angeben, um den symmetrischen Schlüssel zu verschlüsseln.
4. CA Process Automation fügt den verschlüsselten symmetrischen Schlüssel der SOAP-Abfrage im Element <xenc:EncryptedKey> hinzu.
5. Je nach dem Referenztyp, den Sie in "ID-Typ des öffentlichen Schlüssels" angegeben haben, fügt CA Process Automation der SOAP-Anfrage eine Referenz auf den öffentlichen Schlüssel hinzu, die verwendet wird, um den symmetrischen Schlüssel zu verschlüsseln.
6. Der Empfänger entschlüsselt den symmetrischen Schlüssel und verwendet anschließend den entschlüsselten symmetrischen Schlüssel, um die SOAP-Anfrage zu entschlüsseln.

Signatur

1. CA Process Automation verwendet "Alias des privaten Schlüssels" und "Kennwort des privaten Schlüssels", um den privaten Schlüssel aus dem Schlüsselspeicher abzurufen.
2. CA Process Automation verwendet den Kanonisierungsalgorithmus, um die zu verschlüsselnden Daten zu signieren. Diese Daten sind entweder der gesamte Text der Anfrage oder jene Teile, die Sie in "Zu signierende Teile" angeben.
3. CA Process Automation verwendet den privaten Schlüssel im Schlüsselspeicher, um den Inhalt der SOAP-Anfrage mithilfe des Signaturalgorithmus zu signieren, den der Anwender angibt. Das Produkt signiert entweder den gesamten Text der Anfrage oder jene Teile, die Sie in "Zu signierende Teile" angeben.

4. CA Process Automation fügt der SOAP-Anfrage eine Referenz zum Zertifikat bzw. zum öffentlichen Schlüssel, der zum privaten Schlüssel zugeordnet ist, hinzu. Der Anwender gibt den Typ der Referenz in "ID-Typ des öffentlichen Schlüssels" an.
5. Der Empfänger verwendet den öffentlichen Schlüssel, um die Signatur in der SOAP-Abfrage zu validieren.

Fehlerbehebung bei WS-Sicherheit

Die Operatoren "SOAP-Methode aufrufen" und "Asynchrone SOAP-Methode aufrufen" können fehlschlagen, wenn in den folgenden Fällen WS-Sicherheit angewendet wird.

Es kann kein Crypto für die Verwaltung von Schlüsselspeichern erstellt werden

Ein Fehler wird angezeigt, wenn der Operator keinen Crypto erstellen kann, um den Schlüsselspeicher zu verwalten:

```
SOAP invocation failed: Unable to build a Crypto for the keystore containing the Signature/Encryption keys…
```

Dieser Fehler kann unter folgenden Bedingungen auftreten:

- Ungültiges Schlüsselspeicherkennwort
- Ungültiger Schlüsselspeichertyp
- Der Schlüsselspeicherpfad verweist auf eine Datei, die kein Schlüsselspeicher ist.

Überwachen Sie die CA Process Automation-Protokolle im Fehler-Stack, um die Art des Problems zu erhalten. Zum Beispiel:

- Ungültiges Schlüsselspeicherkennwort: Schlüsselspeicher wurde geändert oder das Kennwort ist falsch.
- Ungültiger Schlüsselspeichertyp: java.security.KeyStoreException: *x* wurde nicht gefunden (wobei *x* der angegebene Typ ist)
- Schlüsselspeicherpfad verweist auf eine Datei, die kein Schlüsselspeicher ist: Ungültiges Schlüsselspeicherformat.

Signaturfehler

Bei der Signatur der SOAP-Abfrage können Probleme auftreten, wenn:

- Der Schlüsselspeicherpfad ist nicht auf dem Computer vorhanden, der den Kontaktpunkt enthält.
- Der Schlüsselspeicherpfad fehlt.
- Der Alias des privaten Schlüssels fehlt.
- Sie versuchen einen nicht existierenden Teil der SOAP-Abfrage zu signieren. Das Feld "Grund" enthält eine Meldung:

  `Element to encrypt/sign not found…`
- Der Alias des privaten Schlüssels ist im Schlüsselspeicher nicht vorhanden. Das Feld "Grund" enthält eine Meldung:

  `No certificates for user` *x* `were found for signature…`

  Wobei *x* der angegebene Alias des privaten Schlüssels ist.
- Das Kennwort des privaten Schlüssels ist erforderlich, aber es ist nicht angegeben oder das angegebene Kennwort ist falsch. Der Grund enthält eine Meldung:

  `Cannot recover key…`
- Sie geben einen ungültigen Kanonisierungsalgorithmus an. Das Feld "Grund" enthält eine Meldung:

  `Unknown canonicalizer. No handler installed for URI` *x*`…`

  Wobei *x* der Name des angegebenen Kanonisierungsalgorithmus ist.
- Sie geben einen ungültigen Signaturalgorithmus an. Das Feld "Grund" enthält eine Meldung:

  `The requested algorithm` *x* `does not exist...`

  Wobei *x* der Name des angegebenen Signaturalgorithmus ist.
- Sie geben einen ungültigen ID-Typ des öffentlichen Schlüssels an. Das Feld "Grund" enthält eine Meldung:

  `Unsupported key identification…`

Verschlüsselungsfehler

Bei der Verschlüsselung der SOAP-Abfrage können Probleme auftreten, wenn:

- Der Schlüsselspeicherpfad ist nicht auf dem Computer vorhanden, der den Kontaktpunkt enthält.
- Der Schlüsselspeicherpfad fehlt.
- Sie versuchen einen nicht existierenden Teil der SOAP-Abfrage zu verschlüsseln. Das Feld "Grund" enthält eine Meldung:

  `Element to encrypt/sign not found…`
- Sie versuchen den Algorithmus für symmetrische Verschlüsselung "aes192-cbc" oder "aes256-cbc" zu verwenden, ohne dabei ein Upgrade für die JAR-Dateien der "Unlimited Strength Jurisdiction Policy" durchzuführen. Das Feld "Grund" enthält eine Meldung:

  `Illegal key size or default parameters.`
- Der Alias des öffentlichen Schlüssels ist im Schlüsselspeicher nicht vorhanden. Das Feld "Grund" enthält eine Meldung:

  `No certificates for user` *x* `were found for encryption…`

  Wobei *x* der angegebene Alias des öffentlichen Schlüssels ist.
- Sie geben einen ungültigen Algorithmus für Verschlüsselung an. Das Feld "Grund" enthält eine Meldung:

  `SOAP invocation failed: Unable to encrypt the SOAP message.null.`

  Die WSS4J-Bibliothek löst in diesem Fall einen Fehler im Nullzeiger aus.
- Sie geben einen ungültigen Algorithmus für symmetrische Schlüssel-Verschlüsselung an. Das Feld "Grund" enthält eine Meldung:

  `unsupported key transport encryption algorithm:` *x*

  Wobei *x* der angegebene Algorithmus für symmetrische Schlüssel-Verschlüsselung ist.
- Sie geben einen ungültigen ID-Typ des öffentlichen Schlüssels an. Das Feld "Grund" enthält eine Meldung:

  `Unsupported key identification…`

Hinweis: Wenn Sie einen Teil der SOAP-Abfrage verschlüsseln und anschließend signieren, stellen Sie sicher, den Teil als Inhalt zu verschlüsseln. Dadurch wird sichergestellt, dass "wsu:id" (oder Name und Namespace) *nach* Verschlüsselung und *vor* der Signatur in der SOAP-Abfrage verbleiben. Anderenfalls wird dieser Teil beim Versuch der Signatur nicht gefunden.

Entschlüsselungsfehler

Ein ungültiges Kennwort, dass im Feld "Entschlüsselungskennwort des privaten Schlüssels" angegeben ist, führt dazu, dass der Operator fehlschlägt. Das Feld "Grund" enthält eine Meldung:

```
Unable to apply WSS security on incoming message (SOAP Response).
```

Die Signatur oder Entschlüsselung ist ungültig. Die geschachtelte Ausnahme ist: java.security.

```
UnrecoverableKeyException: Get Key failed:
```

Der angegebene letzte Block ist nicht richtig aufgefüllt.

Ausgabeparameter

serviceURL

methodName

userName

password

httpAuth

soapVersion

inputSource

inlineText

SOAP-Aufrufdaten

serviceURL

Gibt die Service-URL, die der SOAP-Aufruf verwendet, zurück.

methodName

Gibt den Namen der aufzurufenden Methode zurück.

userName

Gibt den Anwendernamen für HTTP-Basisauthentifizierung zurück.

password

Gibt das Kennwort für HTTP-Basisauthentifizierung zurück.

httpAuth

Zeigt je nach Ihrer Angabe für "HTTP-Basisauthentifizierung verwenden?" "wahr" oder "falsch" an.

soapVersion

Gibt die SOAP-Version zurück, die das Produkt für SOAP-Aufrufe verwendet; entweder SOAP 1.1 oder SOAP 1.2.

inputSource

Gibt einen der folgenden Werte für die Eingabequelle des SOAP-Aufrufs zurück:

- InlineText
- Vorformatierte SOAP-Datei
- Ausdruck

inlineText

Enthält die Inline-Textdaten, die für den SOAP-Aufruf verwendet werden. Diese Variable wird nur aufgefüllt, wenn Sie "Inline-Text" als die Eingabequelle auswählen.

Beispiel:

```
<checkServerStatus xmlns="http://www.ca.com/itpam">
CA AuthMinder
<token>token__</token>
<user>user__</user>
<password>password__</password>
</auth>
</checkServerStatus>
```

FileName

Enthält die Inline-Textdaten, die für den SOAP-Aufruf verwendet werden. Diese Variable wird nur aufgefüllt, wenn Sie "Vorformatierte SOAP-Datei" als die Eingabequelle auswählen.

Expression Value

Enthält die Inline-Textdaten, die für den SOAP-Aufruf verwendet werden. Diese Variable wird nur aufgefüllt, wenn Sie "Ausdruck" als die Eingabequelle auswählen.

SavedCallFileName

Gibt den Dateinamen im Feld "Gespeicherte Aufrufdatei" zurück. Diese Datei enthält den SOAP-Umschlag, den das Produkt tatsächlich für SOAP-Aufrufe verwendet.

Beispiel:

```
<?xml version="1.0" encoding="UTF-8"?><SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><SOAP-ENV
:Header/><SOAP-ENV:Body><checkServerStatus
xmlns="http://www.ca.com/itpam">
CA
AuthMinder<token>token__</token><user>user__</user><password>pa
ssword__</password></auth></checkServerStatus></SOAP-ENV:Body><
/SOAP-ENV:Envelope>
```

Dynamische Parameter

Paramsyle

Gibt den Parameterstil zurück, den Sie im Feld "Dynamischer Parameter" ausgewählt haben: "Xpath-Zuweisungen" oder "MacroExpansion".

DynamicParamsInfo

Ein Array von ValueMaps; zeigt die Anzahl von ValueMaps an, die es enthält.

DynamicParamsdata

Gibt Abfrage, Wert und Typ zurück.

Parameter von "Aufrufergebnisse"

responseFileName

Gibt den Dateinamen zurück, den Sie im Feld "Antwortdatei" angegeben haben. Diese Datei enthält die Antwort von einem SOAP-Aufruf.

xPathQuery

Gibt die XPath-Abfrage zurück, die für das Extrahieren der Daten aus den Aufrufergebnissen definiert ist.

datasetVar

Gibt die Variable zurück, die für die extrahierten Aufrufergebnisse erstellt wird.

Typ

Gibt die Variable zurück, die den Datentyp enthält, der vom Anwender für die Aufrufergebnisse definiert wurde.

isExtractToDataSet

Gibt "wahr" zurück, wenn Sie "SOAP-Antworttext zu Datensatzvariable extrahieren" aktivieren. Andernfalls wird "falsch" zurückgegeben.

isExtractHeadersToDataSet

Gibt "wahr" zurück, wenn Sie "SOAP-Header zu Datensatzvariable extrahieren" aktivieren. Andernfalls wird "falsch" zurückgegeben.

isExtractHeadersToIndividualDataSet

Gibt "wahr" zurück, wenn Sie "SOAP-Header-Elemente erster Ebene zu einzelnen Datensatzvariablen extrahieren" aktivieren. Andernfalls wird "falsch" zurückgegeben.

isExtractToIndividualDataSet

Gibt "wahr" zurück, wenn Sie "SOAP-Header-Elemente erster Ebene zu einzelnen Datensatzvariablen extrahieren" aktivieren. Andernfalls wird "falsch" zurückgegeben.

isStripXMLNamespaces

Gibt "wahr" zurück, wenn Sie "XML-Namespaces aus Antwort entfernen" aktivieren. Andernfalls wird "falsch" zurückgegeben.

Betriebsablauf-Ergebnisse

Je nach der Auswahl der Kontrollkästchen in den Parametern "Aufrufergebnisse" enthalten die Betriebsablauf-Ergebnisse die Ausgabe der SOAP-Aufrufergebnisse.

Soap Response Body

Speichert den vollständigen SOAP-Antworttext.

Soap Response Header Data

Speichert den Antwort-Header.

SoapResponseHeader

Enthält die isolierten Daten der SOAP-Header-Antwort.

Soap Response Data

Enthält die isolierten Daten der SOAP-Aufrufergebnisse.

AsyncSoapIntermediateResponse

Enthält die vollständige Antwort mit den durch den SOAP-Aufruf empfangenen Headern.

Beispiel (mithilfe der CheckServerStatus-Methode):

```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<checkServerStatusResponse xmlns="http://www.ca.com/itpam">
<serverStatus>Server status ok.</serverStatus>
</checkServerStatusResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

AsyncSoapInterimResponseBody

Enthält den Text der Antwort des SOAP-Aufrufs.

Beispiel (mithilfe der CheckServerStatus-Methode):

```
<checkServerStatusResponse xmlns="http://www.ca.com/itpam">
<serverStatus>Server status ok.</serverStatus>
</checkServerStatusResponse>
```

AsyncSoapInterimResponseHeader

Enthält den Header der Antwort des SOAP-Aufrufs.

Beispiel (mithilfe der CheckServerStatus-Methode):

```
<SOAP-ENV:Header xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"/>
```

MIME-Anhänge

isResolvedByExpression

Gibt "wahr" zurück, wenn Sie die Option "Ausdruck?" für den MIME-Anhang aktivieren. Andernfalls wird "falsch" zurückgegeben. Durch die Auswahl der Option "Ausdruck?" geben Sie an, dass der MIME-Anhang als Ausdruck angegeben wird, und dass sich dieser Ausdruck auf den MIME-Anhang bezieht.

Expressionfield

Gibt den Ausdruck zurück, der sich auf den MIME-Anhang bezieht.

attachmentFields

Gibt ein ValueMap-Array zurück, das die Anzahl der Elemente enthält, die es umfasst.

Inhaltstyp

Gibt den Inhaltstyp des MIME-Anhangs zurück.

ContentID

Gibt die Inhalts-ID zurück, die den MIME-Anhang eindeutig identifiziert.

FileURL

Gibt den Pfad des MIME-Anhangs zurück.

Kapitel 16: Systemfunktionen von CA Process Automation

Systemfunktionen können für Folgendes verwendet werden:

- Vor- und Nachbearbeitung eines Operators
- Den Quellcode des Operators "JavaScript ausführen".
- ein beliebiges Feld, das einen Ausdruck akzeptiert

In allen diesen Bereichen können auf Systemfunktionen mittels STRG+ALT zugegriffen werden.

Rückgabetypen

Systemfunktionen geben verschiedene Arten von Werten zurück. In einigen wenigen Fällen hängt der Rückgabetyp von den Eingabeparametern oder davon, ob die Funktion "Null" zurückgibt, ab. In den meisten Fällen geben Systemfunktionen Java-Typen und nicht reine JavaScript-Werte zurück. Zum Beispiel geben Systemfunktionen, die laut Beschreibung eine Zeichenfolge zurückzugeben, in Wirklichkeit einen Java.lang.String-Wert zurück. Normalerweise können Sie den Unterschied zwischen Java.lang.String und Zeichenfolge ignorieren. In einigen Fällen wirkt sich dieser Unterschied Verhalten aus, wenn ein Ausdruck oder Skript ausgeführt wird. Im Fall von "typeof x", wobei "x" eine Variable mit dem Rückgabewert einer Systemfunktion ist, ist der Rückgabetyp "Objekt" und nicht "Zeichenfolge".

In der folgenden Tabelle wird die Auswirkung der einzelnen Systemfunktions-Rückgabetypen auf die folgenden Variablen beschrieben:

- Eine JavaScript-Variable wird zum Rückgabewert zugeordnet
- Eine CA Process Automation-Datensatzvariable wird erstellt

Rückgabetyp der Systemfunktion	Typ der JavaScript-Variable nach der Zuordnung	Erstellter Datensatzvariablentyp
Boolescher Wert	Boolescher Wert	Boolescher Wert
C2OString	Objekt	Zeichenfolge
C2OValue	Objekt	ValueMap
C2OValueArray	Objekt	ValueMap
C2OValueMap	Objekt	ValueMap

Rückgabetyp der Systemfunktion	Typ der JavaScript-Variable nach der Zuordnung	Erstellter Datensatzvariablentyp
Datum	Objekt	Datum
int	Nummer	Ganzzahl
Zeichenfolge	Objekt	Zeichenfolge
Zeichenfolge []	Objekt	Zeichenfolge []
void	nicht definiert/-	-/-

absPath

Diese Funktion "absPath" gibt den absoluten Pfad zurück, der mittels Basispfad und relativem Pfad erstellt wurde. Wenn der Basispfad nicht angegeben wird, wird der Basispfad zum aktuellen Prozess verwendet.

Syntax

```
sAbsPath = absPath (path2).
sAbsPath = absPath (path1, path2).
```

Argumente

path1 (Zeichenfolge)

Legt den Basispfad fest.

path2 (Zeichenfolge)

Legt den relativen Pfad fest.

Rückgabewert

sAbsPath (String)

Der absolute Pfad wird als Zeichenfolge zurückgegeben.

Beispiele

1. Es gibt einen Prozess "Process" in einem Ordner "Folder". Um den vollständigen Pfad eines Objekts "Process1" (im gleichen Ordner vorhanden) im Objekt "Process" abzurufen, verwenden Sie absPath('Process1'). In diesem Fall wird der Basispfad automatisch als Basispfad des Objekts "Process" übernommen.

   ```
   sPath = absPath('Process1')
   ```

2. In diesem Fall ist der erste Pfad der beste Pfad des Objekts, der zweite hingegen ist der relative Pfad in Bezug auf den Basispfad. Die Antwort in diesem Fall ist /folder1/process.

   ```
   sPath = absPath("/folder1/folder2","../process");
   ```

adjustDate

Die Funktion "adjustDate" passt ein Datum an eine angegebene Anzahl von Tagen, Wochen, Monaten oder Jahren an.

Syntax

```
newDt = adjustDate(dt, number, type)
```

Argumente

dt (java.util.Date)

Gibt das Datum an, das angepasst werden muss.

number (Int)

Gibt das Datum an, das angepasst werden muss.

type

Gibt den Typ des anzupassenden Werts an. Kann von einem der folgenden Typen sein:

- "y"(year)
- "d"(day)
- "w"(week)
- "m"(month)

Rückgabewert

newDt (java.util.Date)

Das angepasste Datum wird als ein java.util.Date zurückgegeben.

Beispiele

Nehmen Sie an, dass heute der 16.02.2012 ist

1. Das folgende Beispiel gibt als Datum den 18.02.12 zurück, also zwei Tage nach dem aktuellen Datum.

   ```
   adjustDate(today(),2,'d')
   ```

2. Das folgende Beispiel gibt als Datum den 16.02.13 zurück, also ein Jahr nach dem aktuellen Datum.

   ```
   adjustDate(today(),1'y')
   ```

3. Das folgende Beispiel gibt als Datum den 09.02.12 zurück, also eine Woche vor dem aktuellen Datum.

   ```
   adjustDate(today(),-1,'w')
   ```

4. Das folgende Beispiel gibt als Datum den 16.01.12 zurück, also einen Monat vor dem aktuellen Datum.

   ```
   adjustDate(today(),-1,'m')
   ```

adjustResourceVals

Die Funktion "adjustResourceVals" ändert die Werte und den Zustand einer Ressource.

Weitere Informationen über Ressourcen finden Sie unter *Handbuch für Inhaltsdesign*.

Syntax

```
bSuccess =
adjustResourceVals(resPath,resName,freeAmount,total,lock,unlock,re
set)
```

Argumente

resPath (String)

Gibt den Pfad des Ressourcenobjekts an.

resName (String)

Gibt den Namen der Ressource in dem Ressourcenobjekt an, das angepasst werden muss.

freeAmount (Int)

Gibt die Menge freier Ressourcen an.

total (Int)

Der in diesem Argument angegebene Wert ergibt die vorhandene Anzahl der gesamten Ressourcen.

lock (boolean)

Gibt "Wahr" an, wenn die Ressource gesperrt werden muss. Wenn "Sperren" auf "Wahr" gesetzt ist, können die Werte "freeAmount" und "total" für diese Ressource nicht festgelegt werden, und die Systemfunktion gibt "Falsch" zurück.

unlock (Boolean)

Gibt "Wahr" an, wenn die Ressource entsperrt werden muss.

Hinweis: Sie können die gleiche Ressource nicht sperren und entsperren. Wechseln Sie zwischen Sperre und Entsperren, wenn Sie diese Systemfunktion verwenden.

reset (Boolean)

Gibt "Wahr" an, wenn das Ressourcenobjekt zurückgesetzt werden muss.

Hinweis: Das Angeben von "Wahr" hat Vorrang vor allen anderen Operatoren. Alle Parameter der jeweiligen Ressource werden auf die Standardwerte zurückgesetzt (unabhängig von den angegebenen Eingaben in der Systemfunktion).

Rückgabewert

bSuccess (Boolean)

Dies Funktion gibt "wahr" zurück, wenn sie erfolgreich ausgeführt wurde, oder "falsch", wenn sie fehlschlägt.

Beispiel

```
bSuccess =
adjustResourceVals("\Resources","LockResource",20,20,false,true,fa
lse);
```

In diesem Beispiel sind die Variablen folgendermaßen festgelegt:

```
respath = "\Resources"
renName = "LockResource"
freeAmount = 20;
total = 20
lock = false
unlock = true
reset = false
```

applyXPath

Die Funktion "applyXPath" verwendet eine XPath-Abfrage, um XML-Daten zu extrahieren, und gibt ein XML-Fragment mit einer Liste der Knoten zurück, die aus der Anfrage resultieren.

Syntax

```
sXML = applyXPath(xmlData, xpathQuery, namespaceAware, returnArray)
```

Argumente

xmlData (String)

Definiert die XML-Datei, in der die XPath-Anfrage ausgeführt werden soll.

xpathQuery (String)

Definiert die XPath-Anfrage, die ausgeführt werden soll.

namespaceAware (Boolean)

(Optional) Gibt an, Namespaces entfernt werden sollen, bevor die XPath-Anfrage angewendet wird.

Werte:

True: CA Process Automation entfernt keine Namespaces, bevor die XPath-Anfrage angewendet wird.

Falsch: CA Process Automation entfernt Namespaces, bevor die XPath-Anfrage angewendet wird, um die Anfrage zu vereinfachen.

Standard: True

returnArray (Boolesch)

(Optional) Gibt an, ob ein Array mit Ergebnissen oder eine einzelne Zeichenfolge zurückgegeben werden soll.

Werte:

True: Die Funktion gibt ein Array mit Ergebnissen zurück.

False: Die Funktion gibt eine einzelne Zeichenfolge mit mehreren verketteten Ergebnissen zurück.

Standard: False

Hinweis: Wenn die XPath-Anfrage keine Übereinstimmung findet, wird ein leeres Array zurückgegeben.

Rückgabewerte und Beispiele

Einer der folgenden Rückgabewerte, basierend auf dem Wert des Arguments "returnArray".

- Wenn "returnArray" "false" ist:

  sXML (String)

  Wenn Sie das returnArray-Argument weglassen oder auf "false" setzen, gibt die Abfrage diesen Wert zurück. Der Wert, der zurückgegeben wird, ist ein XML-Fragment, in dem die Knoten aufgelistet werden. Zum Beispiel:

  ```
  Process.xpathResult = applyXPath(Process.bookXML,
  "/bookstore/book[2]", true, false);
  ```

- Wenn "returnArray" "true" ist:

  sXMLArray (C2OValueArray)

  Wenn Sie das returnArray-Argument auf "true" setzen, gibt die Anfrage diesen Wert zurück. Der Wert, der zurückgegeben wird, ist ein C2OValueArray-Array, in dem jedes Element ein XML-Fragment für einen Knoten ist. Zum Beispiel:

  ```
  Process.xpathArrayResult = applyXPath(Process.bookXML,
  "/bookstore/book", true, true);
  ```

applyXPathToUrl

Die Funktion "applyXPathToUrl" verwendet eine XPath-Abfrage, um XML-Daten aus einer angegebenen URL zu extrahieren. Die Funktion gibt ein XML-Fragment zurück, das die Knoten auflistet, die sich aus der Anfrage ergeben.

Syntax

```
sXML = applyXPathToUrl(url, xpathQuery, namespaceAware, returnArray)
```

Argumente

url (String)

Definiert die URL des XML-Dokuments, auf dem die XPath-Anfrage ausgeführt werden soll.

xpathQuery (String)

Definiert die XPath-Anfrage, die ausgeführt werden soll.

namespaceAware (Boolean)

(Optional) Gibt an, Namespaces entfernt werden sollen, bevor die XPath-Anfrage angewendet wird.

Werte:

True: CA Process Automation entfernt keine Namespaces, bevor die XPath-Anfrage angewendet wird.

Falsch: CA Process Automation entfernt Namespaces, bevor die XPath-Anfrage angewendet wird, um die Anfrage zu vereinfachen.

Standard: True

returnArray (Boolesch)

(Optional) Gibt an, ob ein Array mit Ergebnissen oder eine einzelne Zeichenfolge zurückgegeben werden soll.

Werte:

True: Die Funktion gibt ein Array mit Ergebnissen zurück.

False: Die Funktion gibt eine einzelne Zeichenfolge mit mehreren verketteten Ergebnissen zurück.

Standard: False

Hinweis: Wenn die XPath-Anfrage keine Übereinstimmung findet, wird ein leeres Array zurückgegeben.

Rückgabewerte und Beispiele

Einer der folgenden Rückgabewerte, basierend auf dem Wert des Arguments "returnArray".

- Wenn "returnArray" "false" ist:

  sXML (String)

  Wenn Sie das Argument "returnArray" weglassen oder auf "false" setzen, gibt die Abfrage ein XML-Fragment zurück, in dem die Knoten aufgelistet sind. Zum Beispiel:

  ```
  sXML =
  applyXPathToUrl("http://localhost:8080/itpam_tutorials/book
  .xml", "/bookstore/book[2]", true, false);
  ```

- Wenn "returnArray" "true" ist:

  sXMLArray (C2OValueArray)

  Wenn Sie das Argument "returnArray" auf "true" setzen, gibt die Abfrage ein C2OValueArray zurück, in dem jedes Element ein XML-Fragment ist, das einem Knoten entspricht. Zum Beispiel:

  ```
  sXMLArray =
  applyXPathToUrl("http://localhost:8080/itpam_tutorials/book
  .xml", "/bookstore/book", true, true);
  ```

checkCalendarDate

Die Funktion "checkCalendarDate" überprüft, ob sich das angegebene Datum innerhalb der angegebenen Bedingung befindet. Die Bedingung umfasst "Include Calendar", "Exclude Calendar", "Delta", "Offene Tage" und "Maximale Aufschübe". Weitere Informationen zu diesen Parametern finden Sie unter der Eigenschaft "Kalender".

Syntax

```
bIsAvail = checkCalendarDate(date, includeCalendar, excludeCalendar,
delta, openDays, maxShifts)
```

Argumente

date (java.util.Date)

Gibt das Eingabedatum an.

includeCalendar (String)

Gibt den Pfad des Objekts des Einschließungskalenders an.

excludeCalendar (String)

Gibt den Pfad des Objekts des Ausschlusskalenders an.

delta (Int)

Gibt das Delta an.

openDays (Int)

Gibt den openDays-Wert an.

maxShifts (Int)

Gibt den maxshifts-Wert an.

Rückgabewert

bIsAvail (Boolean)

Diese Funktion gibt "wahr" zurück, wenn das angegebene Datum offen ist, sonst "falsch".

Beispiel

```
Process.sIncCal=absPath("IncCal");
Process.sExcCal=absPath("ExcCal");Process.bCaldate =
checkCalendarDate(today(),sIncCal,sExcCal,0,false,0);
```

convertJson

Die Funktion "convertJson" konvertiert eine gültige JSON-Zeichenfolge in ValueMap. Verwenden Sie diese Methode, um eine JSON-Antwort von einem REST-Service in ein ValueMap-Objekt zu konvertieren. Das ValueMap-Objekt kann dann mithilfe von standardmäßigen Ausdrücken durchquert und aufgerufen werden.

Syntax

```
vmResult = convertJson(jsonString)
```

Argumente

jsonString (String)

Gibt die zu analysierende JSON-Zeichenfolge an.

Rückgabewert

ValueMap

ValueMap-Darstellung der Daten, die in der JSON-Zeichenfolge enthalten sind, die durchquert oder mit Ausdrücken referenziert werden kann.

Wenn Sie einen Nullwert übergeben, gibt die Methode einen Nullwert ohne Ausnahme zurück. Wenn Sie eine ungültige JSON-Zeichenfolge übergeben, gibt die Methode einen Nullwert zurück, und die Ausnahmen werden auf dem Server protokolliert.

Beispiel

```
vmResult = convertJson(Process.jsonString)
```

Denken Sie an ein Beispiel, bei dem Sie einen REST-Service aufgerufen haben. Die folgende eingehende Antwort wird im restResponse-Wert eines Variablennamens in einem Prozessdatensatz gespeichert:

```
{
    "UserName": "pamadmin",
    "age": 25,
    "address": {
        "streetAddress": "CA Technologies, 115, IT Park Area",
        "city": "Hyderabad",
        "state": "AP",
        "postalCode": "500084"
    },
    "phoneNumber": [
        {
            "type": "Office",
            "number": "04066812345"
        },
        {
            "type": "Home",
            "number": "04066854321"
        }
    ]
}
```

Um auf die Werte von "UserName", "streetAddress" und "phoneNumber" zuzugreifen, schreiben Sie den folgenden Code:

```
// Codestart
// Analysieren Sie die REST-Antwort mit der Methode "convertJson()",
und speichern Sie sie in einer Prozessdatensatzvariablen mit dem Namen
"resultData".
Process.resultData=convertJson(Process.restResponse);
// Greifen Sie über die Variable "resultData" auf "UserName" zu.
Process.userName = Process.resultData.UserName;
// Straßenadresse befindet sich im Adressenobjekt und wird deswegen
mit der folgenden Syntax aufgerufen.
Process.streetAddress = Process.resultData.address.streetAddress;
// Telefonnummer ist ein Array und Arbeitsplatznummer wird im ersten
Element gespeichert, deswegen wird Index [0] verwendet.
Process.officePhoneNumber =
Process.resultData.phoneNumber[0].number;
// Codeende
```

convertValueToXml

Die Funktion "checkCalendarDate" gibt ein XML-Fragment zurück, das auf einem Array oder einer ValueMap mit einfachen Typen basiert.

Bei Verwendung von ValueMaps werden die XML-Elemente mithilfe von Feldnamen als Kennungen und Feldwerten als die Inhalte erstellt. Sie können eine Zeichenfolge oder Null für den Kennungsparameter angeben, wenn Sie ValueMaps verwenden. Wenn Sie eine Zeichenfolge angeben, wird die Zeichenfolge dazu verwendet, ein Stammelement mit der angegebenen Kennung zu erstellen. Die von der ValueMap erstellten Elemente sind in diesem Stammelement enthalten. Wenn Sie Null angeben, sind die Elemente auf der Stammebene.

Bei Verwendung von Arrays müssen Sie eine Zeichenfolge für den Kennungsparameter angeben. Diese Zeichenfolge wird verwendet, um die Elementkennungen mit den Array-Werten als Elementinhalte zu erstellen.

Syntax

```
sXML = convertValueToXml(arrayOrVmap, tag)
```

Argumente

arrayOrVmap (String)

Gibt ein Array oder eine ValueMap an.

tag (String)

Gibt die obligatorische mit einem Array zu verwendende Kennung oder die optionale mit einer ValueMap zu verwendende Kennung an.

Rückgabewert

sXML (String)

Der Rückgabetyp ist "String".

Beispiele

1. In diesem Beispiel enthält Process.array die Werte 1, 2 und 3.

   ```
   Process.xml = convertValueToXml(Process.array, "test")
   ```

   Process.xml enthält das folgende XML-Fragment:

   ```
   <test>1<test><test>2<test><test>3<test>
   ```

2. In diesem Beispiel besitzt Process.valuemap zwei Felder mit den Namen "field1" und "field2" und den Werten "value1" und "value2".

   ```
   Process.xml = convertValueToXml(Process.valuemap, null)
   ```

   Process.xml enthält das folgende XML-Fragment:

   ```
   <field1>value1</field1><field2>value2</field2>
   ```

convertXml

Die Funktion "convertXml" konvertiert ein XML-Fragment in eine ValueMap.

Syntax

```
vmResult = convertXml(xmlString)
```

Argumente

xmlString (String)

Gibt das zu analysierende XML an.

Rückgabewert

vmResult (ValueMap)

Darstellung der in der XML-Zeichenfolge enthaltenen Daten als Wertzuordnung.

Beispiel

```
vmResult = convertXml(Process.xmlString)
```

convertXmlUrl

Die Funktion "convertXmlUrl" konvertiert das XML-Dokument, das über eine URL zugreifbar ist, in eine ValueMap.

Syntax

```
vmResult = convertXmlUrl(url)
```

Argumente

url (String)

Gibt die URL des XML-Dokuments an, die analysiert werden muss.

Rückgabewert

vmResult (ValueMap)

ValueMap-Darstellung der Daten, die aus dem durch die URL angegebenen XML-Dokument abgerufen werden.

Beispiel

```
vmResult = convertXmlUrl("
http://localhost:8080/itpam_tutorials/book.xml");
```

createHyperLink

Die Funktion "createHyperLink" erstellt ein HTML-Hyperlinkelement mit den angegebenen Parametern und gibt eine Zeichenfolge zurück, die als ""+name+"" formatiert ist.

Syntax

```
sLink = createHyperLink(url, name)
```

Argumente

url (String)

Gibt die HTTP-URL an, deren Hyperlink erstellt werden muss.

name (String)

Gibt den Namen des Hyperlinks an.

Rückgabewert

sLink (String)

Ein Hyperlink mit der URL und dem Name, die der Funktion als Argumente übergeben werden.

Beispiel

```
sLink = createHyperLink("http://www.ca.com","CA Technologies");
```

createResourceObject

Die Funktion "createResourceObject" erstellt ein Ressourcenobjekt.

Syntax

```
bSuccess = createResourceObject(resourcePath)
```

Argumente

resourcePath (String)

Gibt den Pfad des Ressourcenobjekts an.

Rückgabewert

bSuccess (Boolean)

Dies Funktion gibt "wahr" zurück, wenn sie erfolgreich ausgeführt wurde, oder "falsch", wenn sie fehlschlägt.

Beispiel

```
bSuccess = createResourceObject("SyncRes")
```

deleteAttachments

Die Funktion "deleteAttachments" löscht Anhänge aus der CA Process Automation-Datenbank, wenn ein Array mit eindeutigen IDs vorhanden ist.

Syntax

```
bSuccess = deleteAttachments(AttachmentIDArray)
```

Argumente

AttachmentIDArray (Array)

Gibt ein Array von eindeutigen IDs an. Der IDs können vom Typ String, Long oder Integer sein.

Rückgabewert

bSuccess (Boolean)

- Dies Funktion gibt nur dann "Falsch" zurück, wenn die Argumente ein leeres Array sind.
- Diese Funktion gibt eine Ausnahme aus (das heißt, der Operator schlägt fehl), wenn sie Anhänge nicht löschen kann oder wenn ungültige Argumente übergeben wurden.
- Dies Funktion gibt "Wahr" zurück, wenn die Anfrage zum Löschen eines Anhangs erfolgreich bearbeitet werden kann (auch wenn die Funktion einige oder alle Anhänge nicht ganz löschen kann).

Beispiele

```
Process.rglAttachIDs = new Array(1,2,3,4,5);
Process.del = deleteAttachments(Process.rglAttachIDs);
```

deleteObject

Die Funktion "deleteObject" löscht und bereinigt das Bibliotheksobjekt, das durch den Parameter "objectName" angegeben wird. Der Eingabeparameter kann ein vollständiger oder relativer Pfad sein. Der relative Pfad ist relativ zu dem Prozess, in dem das Skript ausgeführt wird.

Syntax

```
bSuccess = deleteObject(objectName)
```

Argumente

objectName (String)

Gibt den vollständigen/relativen Pfad des Bibliotheks-Objekts an, das gelöscht und bereinigt werden muss.

Rückgabewert

bSuccess (Boolean)

Dies Funktion gibt "wahr" zurück, wenn sie erfolgreich ausgeführt wurde, oder "falsch", wenn sie fehlschlägt.

Beispiele

1. In diesem Beispiel wird das Prozessobjekt, das sich unter dem Pfad /folder/Process befindet, gelöscht, und der Wert "Wahr" wird zurückgegeben.

   ```
   bSuccess = deleteObject('/folder/Process')
   ```

2. Diese Funktion wird im Prozess "Process_1" ausgeführt, und "Process_1" befindet sich im Ordner "Folder1", der im Stammordner enthalten ist. Diese Funktion löscht den Prozess mit dem vollständigen Pfad "/Folder1/Process", und der Wert "Wahr" wird zurückgegeben.

   ```
   bSuccess = deleteObject('Process')
   ```

deleteResource

Die Funktion "deleteResource" löscht eine Ressource aus einem Ressourcenobjekt.

Syntax

```
bSuccess = deleteResource(resourcePath, resourceName)
```

Argumente

resourcePath (String)

Gibt den Pfad des Ressourcenobjekts an.

resourceName (String)

Gibt den Namen der Ressource in dem Ressourcenobjekt an, die gelöscht werden muss.

Rückgabewert

bSuccess (Boolean)

Dies Funktion gibt "wahr" zurück, wenn sie erfolgreich ausgeführt wurde, oder "falsch", wenn sie fehlschlägt.

Beispiel

```
bSuccess = deleteResource("/folder/ResObject", "fileLock");
```

deleteValueMapField

Die Funktion "deleteValueMapField" löscht ein Feld aus einer ValueMap.

Syntax

```
bSuccess = deleteValueMapField(vMap, fieldName)
```

Argumente

vMap (ValueMap)

Gibt das ValueMap an, dessen Feld gelöscht werden muss.

fieldName (String)

Gibt den Namen des Feldes an, das gelöscht werden muss.

Rückgabewert

bSuccess (Boolean)

Gibt "wahr" zurück, wenn die Löschung erfolgreich war, und andernfalls "falsch".

Beispiel

```
bSuccess = deleteValueMapField(Process.vMap,"price");
```

existsCalendar

Die Funktion "existsCalendar" überprüft, ob unter dem angegebenen Pfad ein Kalenderobjekt vorhanden ist. Der Pfad kann ein absoluter/relativer Pfad sein. Der relative Pfad ist relativ zu dem Prozess, in dem das Skript ausgeführt wird.

Syntax

```
bExists = existsCalendar(calendarPath);
```

Argumente

calendarPath (String)

Gibt den vollen/relativen Pfad des Kalenderobjekts an, dessen Existenz überprüft werden muss.

Rückgabewert

bExists (Boolean)

Gibt wahr zurück, wenn das Kalenderobjekt existiert, sonst falsch.

Beispiel

```
bExists = existsCalendar("WorkCalendar");
```

existsCustomIcon

Die Funktion "existsCustomIcon" überprüft, ob unter dem angegebenen Pfad ein anwenderspezifisches Symbol-Objekt vorhanden ist. Der Pfad kann ein absoluter/relativer Pfad sein. Der relative Pfad ist relativ zu dem Prozess, in dem das Skript ausgeführt wird.

Syntax

```
bExists = existsCustomIcon(customIconPath)
```

Argumente

customIconPath (String)

Gibt den vollen/relativen Pfad des anwenderspezifischen Symbol-Objekts an, dessen Existenz überprüft werden muss.

Rückgabewert

bExists (Boolean)

Gibt "Wahr" zurück, wenn das anwenderspezifische Symbol-Objekt existiert, sonst falsch.

Beispiel

```
if (existsCustomIcon(customIconPath))
{
	Process.mseg_custom_icon= "CustomIcon exists" ;
}
else
{
	Process.mseg_custom_icon= "CustomIcon does not exist" ;
}
```

existsCustomOperator

Die Funktion "existsCustomOperator" überprüft, ob unter dem angegebenen Pfad ein anwenderspezifisches Operator-Objekt vorhanden ist. Der Pfad kann ein absoluter/relativer Pfad sein. Der relative Pfad ist relativ zu dem Prozess, in dem das Skript ausgeführt wird.

Syntax

```
bExists = existsCustomOperator(customOperatorPath)
```

Argumente

customOperatorPath (String)

Gibt den vollen/relativen Pfad des anwenderspezifischen Operators an, dessen Existenz überprüft werden muss.

Rückgabewert

bExists (Boolean)

Gibt "Wahr" zurück, wenn der anwenderspezifische Operator existiert, sonst falsch.

Beispiel

```
if (existsCustomOperator(customOperatorpath))
{
       Process.mseg_custom_operator= "CustomOperator exists" ;
}
else
{
       Process.mseg_custom_icon= "CustomOperator does not exist" ;
}
```

existsDataset

Die Funktion "existsDataset" überprüft, ob unter dem angegebenen Pfad ein Datensatzobjekt vorhanden ist. Der Pfad kann ein absoluter/relativer Pfad sein. Der relative Pfad ist relativ zu dem Prozess, in dem das Skript ausgeführt wird.

Syntax

```
bExists = existsDataset(datasetPath)
```

Argumente

datasetPath (String)

Gibt den vollen/relativen Pfad des Datensatzes an, dessen Existenz überprüft werden muss.

Rückgabewert

bExists (Boolean)

Gibt wahr zurück, wenn das Datensatzobjekt existiert, sonst falsch.

Beispiel

```
if (existsDataset(datasetPath))
{
	Process.mseg_dataset= "Dataset Common exists" ;
}
else
{
	Process.mseg_dataset= "Dataset Common does not exist" ;
}
```

existsFolder

Die Funktion "existsFolder" überprüft, ob unter dem angegebenen Pfad ein Ordnerobjekt vorhanden ist. Der Pfad kann ein absoluter/relativer Pfad sein. Der relative Pfad ist relativ zu dem Prozess, in dem das Skript ausgeführt wird.

Syntax

```
bExists = existsFolder(folderPath)
```

Argumente

folderPath (String)

Gibt den vollen/relativen Pfad des Ordners an, dessen Existenz überprüft werden muss.

Rückgabewert

bExists (Boolean)

Gibt wahr zurück, wenn das Ordnerobjekt existiert, sonst falsch.

Beispiel

```
if(existsFolder(folderpath))
{
Process.mesg_folder = "Folder test exists";
}
else
{
Process.mesg_folder = "Folder test does not exist";
}
```

existsInteractionRequestForm

Die Funktion "existsInteractionRequestForm" überprüft, ob unter dem angegebenen Pfad ein Interaktionsauftragsformular-Objekt vorhanden ist. Der Pfad kann ein absoluter/relativer Pfad sein. Der relative Pfad ist relativ zu dem Prozess, in dem das Skript ausgeführt wird.

Syntax

```
bExists = existsInteractionRequestForm(irfPath)
```

Argumente

irfPath (String)

Gibt den vollen/relativen Pfad des Interaktionsauftragsformular-Objekts an, dessen Existenz überprüft werden muss.

Rückgabewert

bExists (Boolean)

Gibt "wahr" zurück, wenn das Interaktionsauftragsformular-Objekt vorhanden ist, und "falsch", wenn dies nicht der Fall ist.

Beispiel

```
if (existsInteractionRequestForm(irfPath))
{
	Process.mseg_irf= "Interaction Request Form exists" ;
}
else
{
	Process.mseg_irf= "Interaction Request Form does not exist" ;
}
```

existsProcess

Die Funktion "existsProcess" überprüft, ob unter dem angegebenen Pfad ein Prozessobjekt vorhanden ist. Der Pfad kann ein absoluter/relativer Pfad sein. Der relative Pfad ist relativ zu dem Prozess, in dem das Skript ausgeführt wird.

Syntax

```
bExists = existsProcess(processPath)
```

Argumente

processPath (String)

Gibt den vollen/relativen Pfad des Prozessobjekts an, dessen Existenz überprüft werden muss.

Rückgabewert

bExists (Boolean)

Gibt wahr zurück, wenn das Prozessobjekt existiert, sonst falsch.

Beispiel

```
if (existsProcess(processPath))
{
      Process.mseg_process= "\'Pass control to previous oper.\'
exists" ;
}
else
{
      Process.mseg_process= "\'Pass control to previous oper.\' does
not exist" ;
}
```

existsProcessWatch

Die Funktion "existsProcessWatch" überprüft, ob unter dem angegebenen Pfad ein Prozessüberwachungsobjekt vorhanden ist. Der Pfad kann ein absoluter/relativer Pfad sein. Der relative Pfad ist relativ zu dem Prozess, in dem das Skript ausgeführt wird.

Syntax

```
bExists = existsProcessWatch(processWatchPath)
```

Argumente

processWatchPath (String)

Gibt den vollen/relativen Pfad des Prozessüberwachungsobjekts an, dessen Existenz überprüft werden muss.

Rückgabewert

bExists (Boolean)

Gibt wahr zurück, wenn das Prozessüberwachungsobjekt existiert, sonst falsch.

Beispiel

```
if (existsProcessWatch(processWatchPath))
{
	Process.mseg_custom_icon= "ProcessWatch exists" ;
}
else
{
	Process.mseg_process_watch= "ProcessWatch does not exist" ;
}
```

existsResource

Die Funktion "existsResource" überprüft, ob unter dem angegebenen Pfad ein Ressourcenobjekt vorhanden ist. Der Pfad kann ein absoluter/relativer Pfad sein. Der relative Pfad ist relativ zu dem Prozess, in dem das Skript ausgeführt wird.

Syntax

```
bExists = existsResource(resourcePath)
```

Argumente

resourcePath (String)

Gibt den vollen/relativen Pfad des Ressourcenobjekts an, dessen Existenz überprüft werden muss.

Rückgabewert

bExists (Boolean)

Gibt wahr zurück, wenn das Ressourcenobjekt existiert, sonst falsch.

Beispiel

```
bExists = existsResource("/Resources/Locks");
```

existsSchedule

Die Funktion "existsSchedule" überprüft, ob unter dem angegebenen Pfad ein Ablaufplanobjekt vorhanden ist. Der Pfad kann ein absoluter/relativer Pfad sein. Der relative Pfad ist relativ zu dem Prozess, in dem das Skript ausgeführt wird.

Syntax

bExists = existsSchedule(*schedulepath*)

Argumente

schedulepath (String)

Gibt den vollen/relativen Pfad des Ablaufplanobjekts an, dessen Existenz überprüft werden muss.

Rückgabewert

bExists (Boolean)

Gibt "wahr" zurück, wenn das Ablaufplanobjekt vorhanden ist, sonst falsch.

Beispiel

```
if (existsSchedule("testSchedule"))
{Process.mseg_schedule= "testSchedule exists" ;}
else
{Process.mseg_schedule= "testSchedule does not exist" ;}
if (existsSchedule("testSchedule"))
{Process.mseg_schedule= "testSchedule exists" ;}
else
{Process.mseg_Schedule= "testSchedule does not exist" ;}
```

formatDate

Die Funktion "formatDate" gibt eine Zeichenfolge zurück, die auf einem Datums- und einem Formatspezifikator basiert. Zulässige Muster für Datum und Uhrzeit finden Sie in dieser Liste.

Syntax

```
dateString = formatDate(dt, format)
```

Argumente

dt (java.util.Date)

Gibt das zu formatierende Datumsobjekt an.

format (String)

Gibt das erforderliche Format an (zum Beispiel MM/tt/jjjj).

Rückgabewert

dateString (String)

Diese Funktion gibt das Datum als eine Zeichenfolge mithilfe eines Formatspezifikators zurück.

Beispiele

```
Process.logMessage = "Date value is " + formatDate
(Process.CurrentFootprintDate, 'yyyy-mm-dd hh:mm:ss');
```

formatString

Die Funktion "formatString" gibt eine Zeichenfolge zurück, nachdem die angegebene Zeichenfolge mit den angegebenen Argumenten formatiert worden ist.

Syntax

```
resultString = formatString(format, args)
```

Argumente

format (String)

Legt die Formatzeichenfolge fest.

args (Array)

Gibt die Argumente für das Formatieren an.

Rückgabewert

resultString (String)

Diese Funktion gibt eine Zeichenfolge zurück, die entsprechend den in den Funktionsargumenten angegebenen Argumenten formatiert worden ist.

Beispiel

```
var myArray = new Array();
myArray[0] = "a";
myArray[1] = "b";
myArray[2] = "c";
myArray[3] = "d";
sString = formatString("%4$s %3$2s %2$2s %1$2s", myArray);
```

Daraus resultiert sString = "d c b a".

getAllAttachments

Die Funktion "getAllAttachments" gibt die folgenden Informationen für alle Anhänge zurück, die in der CA Process Automation-Datenbank vorhanden sind.

Syntax

```
vmArrayAttachments = getAllAttachments()
```

Rückgabewert

vmArrayAttachments (C2OvalueArray)

Diese Funktion gibt ein Array mit den folgenden Informationen für alle Anhänge, die in der CA Process Automation-Datenbank vorhanden sind, zurück:

contentType

Typ des Anhangsinhalts.

contentID

ID des Anhanginhalts, sofern vorhanden.

fileURL

URL, die verwendet werden kann, um den Anhang anzuzeigen oder herunterzuladen.

name

Name des Anhangs

attachmentID

Eindeutige ID für diesen Anhang. Diese ID kann anderen Systemfunktionen übergeben werden.

Beispiel

```
Process.attachments = getAllAttachments();
```

getAttachmentContent

Die Funktion "getAttachmentContent" ruft den Inhalt aus einem Anhang ab und fügt ihn in eine CA Process Automation-Datensatzvariable ein. Diese Funktion kann Inhalte mit einer maximalen Größe von 64 KB abrufen.

Syntax

```
sAttachment = getAttachmentContent(lAttachmentID)
```

Argumente

lAttachmentID (long)

Die eindeutige ID, die den Anhang identifiziert, in dem sich der abzurufende Inhalt befindet.

Rückgabewert

sAttachment (C2OString)

Der Inhalt wird als C2OString (com.optinuity.c2o.bean.C2OString) zurückgegeben.

Beispiel

```
var i = Process.attachments[0].attachmentID;
Process.cont = getAttachmentContent(i);
```

getCountOfProcessStates

Die Funktion "getCountOfProcessStates" gibt die Anzahl von Prozessen in allen möglichen Zuständen zurück.

Syntax

```
Process.processStates = getCountOfProcessStates();
```

Rückgabewert

processStates (ValueMap)

Gibt eine ValueMap zurück, die aus allen Zuständen und der Anzahl der Prozesse in diesem Zustand besteht. Wenn keine Prozesse im angegebenen Zustand sind, ist die Anzahl 0.

Beispiel

```
Process.processStates = getCountOfProcessStates();
```

getEEMArtifactToken

Diese Funktion generiert einen CA EEM-getEEMArtifactToken, normalerweise zur einmaligen Verwendung.

Syntax

```
getEEMArtifactToken (certificateFilePath, {certificatePassword |
KeyFilePath})
```

Argumente

certificateFilePath

Der relative Pfad (Dateipfad) zur Zertifikatsdatei. Diese Zertifikatsdatei wird mithilfe der Option "Anwenderressourcen verwalten" innerhalb der Registerkarte "Konfiguration" hochgeladen.

certificatePassword (String) | KeyFilePath (String)

Eines der folgenden Argumente:

- Wenn FIPS in CA EEM unterstützt wird, legen Sie dieses Argument auf den relativen Pfad der Schlüsseldatei fest (KeyFilePath). Diese Schlüsseldatei wird mithilfe der Option "Anwenderressourcen verwalten" innerhalb der Registerkarte "Konfiguration" in CA Process Automation hochgeladen.
- Wenn FIPS in CA EEM nicht unterstützt wird, legen Sie dieses Argument auf das Zertifikatskennwort fest. Dieses Kennwort wird als eine Zeichenfolge für das Zertifikat verwendet, das als Argument "certificateFilePath" bezeichnet wird.

Rückgabewert

CA EEM-Artefakt-Token (Zeichenfolge)

Beispiele

Das folgende Beispiel zeigt, dass FIPS-Support in CA EEM aktiviert ist:

```
Process.artifactToken = getEEMArtifactToken
(".c2ouserresources/mycerts/pam.12", "mypassword")
Process.artifactToken = getEEMArtifactToken
(".c2ouserresources/mycerts/upload/pam.cer",
".c2ouserresources/mycerts/keys/pam.key")
```

getEEMArtifactTokenForUser

Die Funktion "getEEMArtifactTokenForUser" generiert ein CA EEM-Token zur einmaligen Verwendung.

Syntax

```
getEEMArtifactTokenForUser (username , password)
```

Argumente

username

Der Anwendername für einen CA EEM-Anwender.

password

Das Kenntwort für einen CA EEM-Anwender.

Rückgabewert

CA EEM-Artefakt-Token (Zeichenfolge)

Beispiel

```
Process.artifactToken = getEEMArtifactTokenForUser ("pamadmin",
"pamadmin")
```

getEEMCredentialsToken

Die Funktion "getEEMCredentialsToken" generiert ein Token der CA EEM-Anmeldeinformationen, normalerweise für mehrere Verwendungen. Das Argument "certificateFilePath" erwartet einen relativen Pfad (Dateipfad) der Zertifikatsdatei. Diese Datei wird mithilfe von "Anwenderressourcen verwalten" innerhalb der Registerkarte "Konfiguration" hochgeladen.

Wenn FIPS nicht aktiviert ist, dann ist das zweite Argument das Zertifikatskennwort. Dieses Kennwort wird als Zeichenfolge für das Zertifikat verwendet, das innerhalb des ersten Arguments (certificateFilePath) zugeteilt ist.

Wenn FIPS-Unterstützung innerhalb von CA EEM aktiviert ist, dann ist das zweite Argument "KeyFilePath". "KeyFilePath" ist der relative Pfad der Schlüsseldatei. Diese Datei wird mithilfe von "Anwenderressourcen verwalten" auf CA Process Automation hochgeladen.

Syntax

```
getEEMCredentialsToken (certificateFilePath, {certificatePassword |
KeyFilePath})
```

Argumente

certificateFilePath

Der relative Pfad der Zertifikatsdatei.

certificatePassword (String) | KeyFilePath (String)

Eines der folgenden Argumente:

- Bei Nicht-FIPS-Modus sollte dieses Argument das Zertifikatskennwort sein.
- Bei FIPS-aktiviertem Modus sollte dieses Argument das Zertifikat "KeyFilePath" sein. "KeyFilePath" ist eine Datei, die mithilfe von "Anwenderressourcen verwalten" auf CA Process Automation hochgeladen wird.

Rückgabewert

CA EEM-Artefakt-Token (Zeichenfolge)

Beispiel

Bevor Sie den Code schreiben, überprüfen Sie zuerst, ob CA EEM FIPS-aktiviert ist oder nicht, indem Sie die Funktion isFIPSMode (siehe Seite 594) () verwenden, sodass Sie "certificateFilePath", "certificatePassword" oder "certificatepath" und "KeyFilePath" übergeben können.

Zum Beispiel:

```
If (isFIPSMode ().equals('true')){
Process.credentialToken =
getEEMCredentialsToken("/mycerts/upload/pam.cer",
"/mycerts/keys/pam.key")
} else {
Process.credentialToken = getEEMCredentialsToken("/mycerts/pam.12",
"mypassword")
}
```

getEEMCredentialsTokenForUser

Die Funktion "getEEMCredentialsTokenForUser" generiert ein CA EEM-Token zur mehrmaligen Verwendung.

Syntax

```
getEEMCredentialsTokenForUser (username , password)
```

Argumente

username

Der Anwendername für einen CA EEM-Anwender.

password

Das Kenntwort für einen CA EEM-Anwender.

Rückgabewert

CA EEM-Artefakt-Token (Zeichenfolge)

Beispiel

```
Process.credentialToken =
getEEMCredentialsTokenForUser("pamadmin","pamadmin")
```

getEnvVar

Die Funktion "getEnvVar" gibt den Wert einer Umgebungsvariablen von der (BS-)Umgebung zurück, die vor dem Start von JVM festgelegt worden sein muss.

Syntax

```
sEnvValue = getEnvVar(vname)
```

Argumente

vname (String)

Gibt den Namen der Umgebungsvariable an, deren Wert benötigt wird.

Rückgabewert

sEnvValue (String)

Der Wert der angegebenen Umgebungsvariablen in Form einer Zeichenfolge.

Beispiel

```
Process.username=getEnvVar("username");
Process.path = getEnvVar("path");
```

getOrchestratorURL

Die Funktion "getOrchestratorURL" gibt den Namen des Koordinationsrechners zurück.

Syntax

```
getOrchestratorURL()
```

Rückgabewert

URL des Koordinationsrechners (Zeichenfolge)

Gibt die URL des Koordinationsrechners oder des Lastenausgleichs (im Fall eines Clusters) an.

Beispiel

```
Process.x = getOrchestratorURL();
```

In diesem Fall ist *x* der Prozessname. Sobald der Prozess ausgeführt wird, enthält die Variable x im Prozessdatensatz den Namen des Koordinationsrechners.

getPartialAttachmentContent

Die Funktion "getPartialAttachmentContent" wird verwendet, um den Inhalt aus einem Anhang abzurufen. Diese Funktion ist auf den Abruf von maximal 64 KB Inhaltsdaten beschränkt. Sie soll es ermöglichen, eine Teilmenge des Inhalts abzurufen.

Syntax

```
sContent = getPartialAttachmentContent(lAttachmentID,nStartIndex,
nRetrieveLength)
```

Argumente

lAttachmentID (long)

Die eindeutige ID, die den Anhang identifiziert, in dem sich der abzurufende Inhalt befindet.

nStartIndex (Int)

Der Speicherort in Byte

nRetrieveLength (Int)

Rückgabewert

sContent (C2OString)

Enthält den Inhalt des angegebenen Anhangs, der an der Position nStartIndex Byte beginnt und maximal nRetrieveLength Byte umfasst. Der Rückgabetyp ist C2OString (com.optinuity.c2o.bean.C2OString).

Beispiel

```
var  i = Process.attachments[0].attachmentID;
Process.part=getPartialAttachmentContent(i,i+1.100000);
```

getResourceAvail

Die Funktion "getResourceAvail" gibt den Wert des freien Felds einer Ressource in einem Ressourcenobjekt zurück.

Syntax

```
nAvail = getResourceAvail(resourcePath, resourceName)
```

Argumente

resourcePath (String)

Gibt den Pfad des Ressourcenobjekts an.

resourceName (String)

Gibt den Namen der Ressource in dem Ressourcenobjekt an, dessen freies Feld zurückgegeben werden soll.

Rückgabewert

nAvail (Int)

Diese Funktion gibt die Anzahl verfügbarer Ressourcen von der angegebenen Ressource im Ressourcenobjekt im angegebenen Pfad zurück.

Beispiel

```
nAvail = getResourceAvail("/Resources/Locks", "DiskLock");
```

getResourceName

Die Funktion "getResourceName" gibt ein Array von Zeichenfolgen zurück, das die Namen der Ressourcen im Ressourcenobjekt enthält.

Syntax

```
resourceNames = getResourceNames(resourcePath)
```

Argumente

resourcePath (String)

Gibt den Pfad des Ressourcenobjekts an.

Rückgabewert

resourceNames (String Array)

Array von Ressourcennamen, die im angegebenen Ressourcenobjekt enthalten sind.

Beispiel

```
resourceNames = getResourceNames("/Resources/Locks");
```

getResourceTotal

Die Funktion "getResourceTotal" gibt die Gesamtmenge einer bestimmten Ressource in einem Ressourcenautomatisierungsobjekt zurück.

Syntax

```
nTotal = getResourceTotal(resourcePath, resourceName)
```

Argumente

resourcePath (String)

Gibt den Pfad des Ressourcenobjekts an.

resourceName (String)

Gibt den Namen der Ressource in dem Ressourcenobjekt an, dessen Gesamtmenge zurückgegeben werden soll.

Rückgabewert

nTotal (Int)

Gibt die Gesamtanzahl für die angegebene Ressource zurück. Wenn die Ressource nicht vorhanden ist, wird "-1" zurückgegeben.

Beispiel

```
nTotal = getResourceTotal("/Resources/Locks", "DiskLock");
```

getTouchpoints

Die Funktion "getTouchpoints" gibt eine Liste von Kontaktpunkten zurück, die von einen Kontaktpunktnamen oder einem Kontaktpunktgruppennamen referenziert werden. Diese Methode gibt ein Array von Zeichenfolgen zurück. Ein leeres Array heißt, dass die Kontaktpunktgruppe leer ist oder kein Kontaktpunkt mit dem angegebenen Namen existiert.

Syntax

```
rgsTouchpoints = getTouchpoints(TouchpointOrGroupName)
```

Argumente

TouchpointOrGroupName (String)

Gibt den Namen des Kontaktpunkts oder der Kontaktpunktgruppe an.

Rückgabewert

rgsTouchpoints (String Array)

Ein Array von Zeichenfolgen mit Kontaktpunktnamen.

Beispiel

```
rgsTouchpoints = getTouchpoints("localhost");
```

getValueFromValueMapArray()

Die Funktion "getValueFromValueMapArray" gibt ein Feld oder eine Spalte aus einem Array zurück, das auf den angegebenen Parametern basiert.

Syntax

```
getValueFromValueMapArray(groupName, arrName, fieldName,
fieldValue, requiredFieldName)
```

Argumente

groupName (String)

Definiert den Namen der anwenderspezifischen Gruppe, die Sie veröffentlichen.

arrName (String)

Definiert den Namen eines C2OValueMap-Arrays.

fieldName (String)

Definiert den Namen des Feldes, für den die Werte von anderen Spalten oder Feldern in einem Array abgerufen werden sollen.

fieldValue (String)

Definiert den Istwert des Feldes, für den Sie die Werte von anderen Spalten oder Feldern in einem Array abrufen.

requiredFieldName (String)

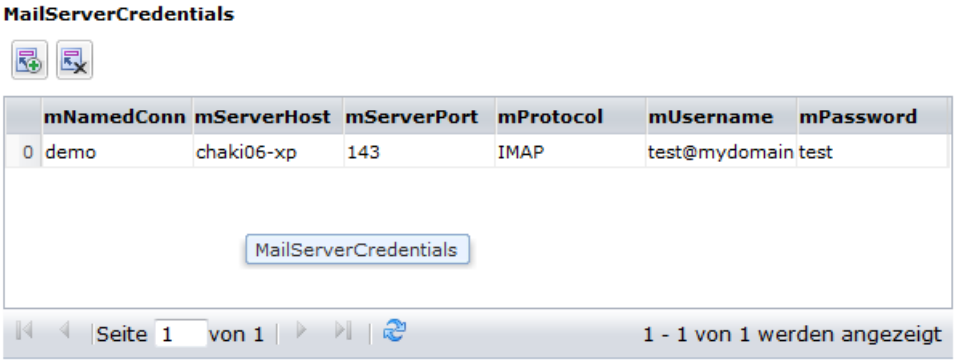
Definiert den Namen des Feldes oder der Spalte, die Sie benötigen. Zum Beispiel: Anwendername, Kennwort oder URL.

Rückgabewert

Der Rückgabewert ist ein C2OValue-Rückgabetyp.

Beispiel

Beachten Sie die Gruppe mit dem Namen "DemoGroup" und ein Array, wie in der folgenden Abbildung dargestellt:



Folgende Syntax ruft den Wert des Feldes "mUsername" aus dem Array für die Verbindung mit dem Namen "DemoGroup" ab:

```
getValueFromValueMapArray("DemoGroup", "MailServerCredentials",
"mNamedConnection", "demo", "mUsername")
```

Die Ausgabe ist ein C2OValue, wobei:

- Der Wert für "fieldName" ist "mUsername"
- Der Wert für "fieldValue" ist "test@mydomain.com"

getValueMapFields

Die Funktion "getValueMapFields" gibt die Liste von Feldnamen in einer ValueMap als Array von Zeichenfolgenvariablen zurück.

Syntax

```
fieldNames = getValueMapFields(vmap)
```

Argumente

vmap (ValueMap)

Gibt das ValueMap-Objekt an, dessen Feldnamen zurückgegeben werden müssen.

Rückgabewert

fieldNames (String Array)

Ein Array von Zeichenfolgen, das die Namen der im ValueMap gefundenen Felder enthält.

Beispiel

```
fieldNames = getValueMapFields(Process.vmBooks);
```

getValuesFromValueMapArray()

"getValuesFromValueMapArray" gibt alle Spalten oder Felder aus einem Array zurück.

Syntax

```
getValuesFromValueMapArray(groupName, arrName, fieldName,
fieldValue)
```

Argumente

groupName (String)

Definiert den Namen der anwenderspezifischen Gruppe, die Sie veröffentlichen.

arrName (String)

Definiert den Namen eines Arrays von "C2OValueMap".

fieldName (String)

Definiert den Namen des Feldes, gegen den der Wert von anderen Spalten oder Feldern in einem Array abgerufen werden soll.

fieldValue (String)

Definiert den Istwert des Feldes, gegen den Sie den Wert von anderen Spalten oder Feldern in einem Array abrufen.

Rückgabewerte

Der Rückgabewert ist ein C2OValue-Rückgabetyp.

hasField

Die Funktion "hasField" bestimmt, ob in einer ValueMap ein Feld vorhanden ist.

Syntax

```
bHasField = hasField(valuemap, fieldName)
```

Argumente

valuemap (ValueMap)

Gibt das zu überprüfende ValueMap-Objekt an.

fieldName (String)

Gibt den Feldnamen an.

Rückgabewert

bHasField (Boolean)

Gibt "wahr" zurück, wenn das Feld in der ValueMap vorhanden ist, oder andernfalls "falsch".

Beispiel

```
bHasField = hasField(Process.vmBooks, "author");
```

include

Die Funktion "include" lädt JavaScript-Code, der in einer Datei für die Verwendung im Vor- oder Nachbearbeitungsabschnitt jedes Operators oder im SourceCode-Abschnitt (siehe Seite 438) des Operators "JavaScript ausführen" definiert ist. Der geladene JavaScript-Code ist nur für die Dauer des Vorbearbeitungsabschnitts, Nachbearbeitungsabschnitts oder SourceCode-Abschnitts, in den er geladen wird, gültig.

Syntax

```
include(jsFile)
```

Argumente

jsFile (String)

Dieser Parameter muss sich auf eine JavaScript-Datei beziehen. Dieser Parameter kann eine HTTP-URL sein. In diesem Fall muss der Parameter mit einer Datei-URL beginnen, oder es kann ein Pfad im Ordner "c2ouserresources" sein. Der Pfad muss relativ zum Ordner ".c2ouserresources" sein. ".c2ouserresources" befindet sich im Ordner ".c2orepository" im CA Process Automation-Installationsordner.

Rückgabewert

Es wird kein Wert zurückgegeben (void).

Beispiele

Das folgende Beispiel ist eine HTTP-URL. Der Parameter fängt mit einer Datei-URL an.

```
include('http://test.ca.com/test.js')
```

Die folgenden Beispiele laden die Testdatei von einem relativen Pfad in den Ordner ".c2ouserresources".

```
include('file://c:/test.js')
Include('test.js')
```

isFIPSMode

Mit der Funktion "isFIPSMode" können Sie programmatisch feststellen, ob der CA EEM-Server den FIPS-Modus aktiviert hat. Diese Funktion gibt "Wahr" zurück, wenn der CA EEM-Server mit aktiviertem FIPS-Modus ausgeführt wird, und "Falsch" wird zurückgegeben, wenn der Server nicht ausgeführt wird.

Syntax

```
isFIPSMode()
```

Rückgabewert

Wahr

Dieser Wert wird zurückgegeben, wenn der CA EEM-Server mit aktiviertem FIPS-Modus ausgeführt wird. (Zeichenfolge)

False

Dieser Wert wird zurückgegeben, wenn der CA EEM-Server mit deaktiviertem FIPS-Modus ausgeführt wird. (Zeichenfolge)

isTouchpointUp

Die Funktion "isTouchpointUp" bestimmt, ob ein Kontaktpunkt aktiv ist.

Syntax

```
bIsUp = isTouchpointUp(touchpointName)
```

Argumente

touchpointName (String)

Gibt den Namen des Kontaktpunkts an.

Rückgabewert

bIsUP (Boolean)

Gibt "wahr" zurück, wenn der Kontaktpunkt aktiv ist, oder andernfalls "falsch".

Beispiel

```
bIsUp = isTouchpointUp("AccountingTouch");
```

load

Die Funktion "load" lädt JavaScript-Code, der in einer Datensatzvariablen für die Verwendung im Vorbearbeitungs- oder Nachbearbeitungsabschnitt jedes Operators oder im SourceCode-Abschnitt (siehe Seite 438) des Operators "JavaScript ausführen" definiert ist. Der geladene JavaScript-Code ist nur für die Dauer des Vorbearbeitungs-, Nachbearbeitungs- oder SourceCode-Abschnitts, in den er geladen wird, gültig.

Syntax

`load(`*`jsCode`*`)`

Argumente

jsCode (String)

Gibt den zu ladenden JavaScript-Code an.

Rückgabewert

Es wird kein Wert zurückgegeben (void).

Beispiele

1. Definieren Sie ein Datensatzobjekt mit dem Namen "Common" und einem Parameter "jsCode", der das folgende JavaScript enthält:

   ```
   function convertToUpperCase(sValue) {return
   sValue.toUpperCase()};
   ```

2. Verwenden Sie die Ladefunktion, um dieses Code-Stück zu laden und Funktionen in diesem Code für Sie verfügbar zu machen:

   ```
   load(Datasets["Common"].jsCode);
   Process.ucValue = convertToUpperCase("helloworld");
   ```

lockResource

Die Funktion "lockResource" sperrt oder entsperrt eine oder mehrere Ressourcen in einem Ressourcenobjekt. Wenn Sie einen Wert für "resourceName" angeben, setzt die Funktion nur diese Ressource zurück. Wenn Sie "resourceName" leer lassen, setzt sie alle Ressourcen im Ressourcenobjekt zurück.

Hinweise:

- Wenn die nicht vorhandenen Ressourcen als Eingabe angegeben werden, werden Ressourcen trotzdem erstellt.
- Wenn Sie einen Wert für "resourceName" angeben und diese Ressource nicht existiert, wird die Ressource mit einer Menge von null erstellt und auf den angegebenen Zustand festgelegt.

Syntax

```
bSuccess = lockResource(resourcePath, resourceName, state)
```

Argumente

resourcePath (String)

Gibt den Pfad des Ressourcenobjekts an.

resourceName (String)

Gibt den Namen der Ressource in dem Ressourcenobjekt an.

state (Boolean)

Gibt an, ob die Ressource gesperrt oder entsperrt werden sollte. Bei gesperrt auf "wahr" festgelegt, bei entsperrt auf "falsch" festgelegt.

Rückgabewert

bSuccess (Boolean)

Gibt "Wahr" zurück, wenn die Funktion erfolgreich ausgeführt wird, oder "Falsch", wenn ihre Ausführung fehlschlägt.

Beispiel

```
bSuccess = lockResource("/Resources/Locks", "InvLock", true);
```

logEvent

Die Funktion "logEvent" fügt eine anwenderspezifische Meldung in die Protokolle einer Prozessinstanz ein.

Syntax

```
logEvent(level, category, msg)
```

Argumente

level (Int)

Gibt eine der folgenden Protokollebenen an:

4 = Error (Fehler)

3 = Warning (Warnung)

2 = Notice (Hinweis)

1 = Normal (Erwartet)

category (Zeichenfolge) (dieser Wert ist optional)

Gibt eine der folgenden Protokollkategorien an:

- "CUSTOM" (Standardeinstellung)
- "FLOW_CATEGORY"
- "AGENDA_CATEGORY"
- "ICON_CATEGORY"
- "HANDLERS_CATEGORY"
- "RESPONSE_CATEGORY"
- "OTHERS_CATEGORY" (PROCESS,AGENDA,OPERATOR,HANDLER, RESPONSE,OTHERS, CUSTOM,OTHERS)

msg (String)

Gibt die Protokollmeldung an.

Rückgabewerte

Es wird kein Wert zurückgegeben (void).

Beispiel

```
logEvent(1, "FLOW_CATEGORY", "Start New Hire Process has completed");
```

newValueMap

Die Funktion "newValueMap" erstellt eine neue ValueMap und gibt sie zurück.

Syntax

```
vmData = newValueMap()
```

Argumente

Keine.

Rückgabewert

vmData (ValueMap)

Beispiel

```
Process.myVmap = newValueMap()
```

newDataset

Durch die Funktion "newDataset" wird ein neues ValueMap, das einem Datensatzobjekt zugewiesen werden kann, dynamisch erstellt und zurückgegeben. Wenn Sie einen Datensatz erstellen, der bereits mit dieser Funktion vorhanden ist, dann ist dieser vorhandene Datensatz nicht betroffen.

Syntax

```
Datasets["/mydataset"]=newDataset()
```

Argumente

Keine.

Rückgabewert

```
ValueMap
```

Beispiel

```
Datasets["/mydataset"]=newDataset()
```

nextOpenDate

Die Funktion "nextOpenDate" gibt ein Öffnungsdatum mit angegebenem "targetDate" unter Berücksichtigung von "includeCalendar", "excludeCalendar" und "maxShifts" zurück. Wenn kein offenes Datum mit den angegebenen Eingaben gefunden wird, ist das Ergebnis null.

Syntax

```
dtNextOpenDate = nextOpenDate(targetDate, includeCalendar,
excludeCalendar, maxshift)
```

Argumente

targetDate (java.util.Date)

Gibt das gewünschte Datum an.

include_calendar (String)

Gibt den Pfad des Objekts des Einschließungskalenders an.

exclude_calendar (String)

Gibt den Pfad des Objekts des Ausschlusskalenders an.

maxshifts (Int)

Gibt bei der Suche nach einem offenen Datum die größtmögliche annehmbare Anzahl von Aufschüben an. Geben Sie positive Zahlen an, um das Datum zu erhöhen, und negative Zahlen, um das Datum zu verringern. Das System beschränkt die Höchstanzahl von Aufschüben auf 5.

Rückgabewert

dtNextOpenDate (java.util.Date)

now

Die Funktion "now" gibt das aktuelle Datum einschließlich der Uhrzeit zurück.

Syntax

```
dtNow = now()
```

Argumente

Keine.

Rückgabewert

dtNow (java.util.Date)

Der Rückgabewert besteht aus dem aktuellen Datum und der aktuellen Uhrzeit.

parseDate

Die Funktion "parseDate" gibt ein Datumsobjekt zurück, nachdem die angegebene Zeichenfolge im erforderlichen Format analysiert wurde.

Syntax

```
dtDate = parseDate(dateStr, format)
```

Argumente

dateStr (String)

Gibt die Zeichenfolge an, die als Datum analysiert werden muss.

format (String)

Gibt das Format an, das zur Interpretation der angegebenen Datumszeichenfolge erforderlich ist, zum Beispiel: MM/tt/jjjj.

Rückgabewert

dtDate (Date)

Das Datumsäquivalent der Eingabe "dateStr String" gemäß der Analyse mit dem Eingabeformat "String".

Beispiele

1. Legt "dtDate" auf "Dec 10, 2009 12:00:00 AM" fest, wenn das Format MM/dd/yyyy ist.
   ```
   dtDate = parseDate('12/10/2009','MM/dd/yyyy')
   ```
2. Legt "dtDate" auf "Oct12, 2009 12:00:00 AM" fest, wenn das Format dd/MM/yyyy ist.
   ```
   dtDate = parseDate('12/10/2009','dd/MM/yyyy')
   ```
3. Schlägt für die folgende Eingabe fehl, da die eingegebene Datumszeichenfolge nicht mit dem Eingabeformat übereinstimmt.
   ```
   dtDate = parseDate('12-10-/2009','dd/MM/yyyy')
   ```

resetResource

Die Funktion "resetResource" setzt eine oder mehrere Ressourcen in einem Ressourcenobjekt zurück, indem sie sie entsperrt und die Anzahl der Verwendungen auf null setzt. Wenn Sie einen Wert für "resourceName" angeben, wird nur diese Ressource zurückgesetzt. Wenn Sie "resourceName" leer lassen, werden alle Ressourcen im Ressourcenobjekt zurückgesetzt.

Hinweis: Wenn Sie einen Wert für "resourceName" angeben und diese Ressource nicht existiert, wird die Ressource mit einer Menge von null erstellt und auf den entsperrten Zustand festgelegt.

Syntax

```
bSuccess = resetResource(resourcePath, resourceName)
```

Argumente

resourcePath (String)

Gibt den Pfad des Ressourcenobjekts an.

resourceName (String)

Gibt den Namen der Ressource in dem Ressourcenobjekt an.

Rückgabewert

bSuccess (Boolean)

Gibt "Wahr" zurück, wenn die Funktion erfolgreich ausgeführt wird, oder "Falsch", wenn ihre Ausführung fehlschlägt.

rollDate

Die Funktion "rollDate" wird verwendet, um einen bestimmten Wert, der auf einem Datum basiert, zu verschieben. Beispielsweise möchten Sie möglicherweise eine Feedback-E-Mail senden, und zwar einen Tag nachdem die Service Desk-Anforderung geschlossen wurde. Verwenden Sie in diesem Fall zum Schreiben der Automatisierungslogik diese Funktion.

Wenn diese Funktion ausgeführt wird, wird der Wert "num" anhand des Typs dem Datum "dt" hinzugefügt.

Syntax

```
dtRollDate = rollDate(dt, num, type)
```

Argumente

dt (java.util.Date)

Gibt das Datumsobjekt an, das Grundlage der erforderlichen Verschiebung sein soll.

num (Int)

Gibt den Wert an, der verschoben werden muss.

type (String)

Gibt den einen der folgenden Werte an:

- "y"(year)
- "d"(day)
- "w"(week)
- "m"(month)

Rückgabewert

dtRollDate (java.util.Date)

Beispiele

1. Gibt das Datum zurück, das zwei Tage in der Zukunft liegt.

   ```
   dtRollDate = rollDate(today(),2,'d')
   ```

2. Gibt das aktuelle Datum im nächsten Jahr zurück, wenn heute z. B. 12. November 2009 ist, wird in diesem Beispiel 1. Januar 2010 zurückgegeben.

   ```
   dtRollDate = rollDate(today(),1 'y')
   ```

3. Gibt den ersten Tag der Vorwoche zurück.

   ```
   dtRollDate = rollDate(today(), -1,'w')
   ```

4. Gibt den ersten Tag des Vormonats zurück.

   ```
   dtRollDate = rollDate(today(),-1 ,'m')
   ```

rollTime

Die Funktion "rollTime" verschiebt die aktuelle Stundenangabe auf den angegebenen Wert und gibt den Stundenwert im 24-Stunden-Format zurück. In dieser Uhrzeit-Konvention dauert der Tag von Mitternacht zu Mitternacht und wird in 24 Stunden unterteilt, die von 0 bis 23 nummeriert werden.

Syntax

```
nHTime = rollTime(num, type)
```

Argumente

num (Int)

Gibt den Wert an, der verschoben werden muss.

type (String)

Dieser Wert kann nur "h" lauten.

Rückgabewert

nHTime (Int)

Beispiel

Gibt die Uhrzeit zurück, die drei Stunden vor der aktuellen Uhrzeit liegt. Wenn derzeit beispielsweise 9 PM ist, gibt dieses Beispiel 18 zurück, was im 24-Stunden-Format 6 PM entspricht.

```
nHTime = rollTime(-3 ,'h')
```

saveAttachmentToFile

Die Funktion "saveAttachmentToFile" speichert den Inhalt eines Anhangs, der durch eine eindeutige ID identifiziert wird, am angegebenen Dateispeicherort. Die Funktion gibt den absoluten Pfad der neuen mit dem Anhangsinhalt erstellten Datei zurück.

Syntax

```
sFileName = saveAttachmentToFile(nAttachmentID, sFileDirName)
```

Argumente

lAttachmentID (long)

Gibt eine eindeutige ID an, die den Anhang identifiziert, der den gewünschten Inhalt enthält.

sFileDirName (String)

Vollständiger Pfad und Dateiname des Speicherorts, an dem die Datei erstellt werden soll.

Wenn kein Dateipfad angegeben ist, wird die Datei im Verzeichnis <*Installationsverzeichnis*>/server/c2o erstellt.

Wenn keine Datei angegeben wird, wird ein eindeutiger Dateiname generiert.

Wenn nur ein Pfad angegeben wird, muss die Pfadangabe durch ein Pfadtrennzeichen ('\' für Windows oder '/' für UNIX) abgeschlossen werden.

Rückgabewert

sFileName (C2OString)

Der vollständige Pfad der Datei, einschließlich des Dateinamens, wird zurückgegeben, wenn die Funktion erfolgreich ist. Wenn die Funktion fehlschlägt, wird NULL zurückgegeben. Der Rückgabetyp ist "C2OString".

Beispiel

```
var  i = Process.attachments[0].attachmentID;
Process.save =  saveAttachmentToFile( i, "attach.txt");
```

setOperatorStatus

Die Funktion "setOperatorStatus" wird verwendet, um einen Fehler oder eine Übergabe des Parameters zu erzwingen.

Syntax

```
setOperatorStatus (operatorStatus, operationResult, reason)
```

Argumente

Operator Status (String)

Gibt den Zustand des Operators an. Dieses Argument kann nur entweder Erfolgs- oder Fehlerwerte übernehmen.

Operation Result (Int)

Gibt das Betriebsablaufergebnis an. Dieses Argument überschreibt das Operatorergebnis im Operatordatensatz.

Reason (String)

Gibt den Grund an, der den Operatorgrund im Operatordatensatz überschreibt.

Rückgabewert

Es wird kein Wert zurückgegeben (void).

Beispiel

Im folgenden Beispiel wird der Erfolg des Operators erzwungen, wobei das Operatorergebnis 1 und der Grund "force success" ist.

```
setOperatorStatus("success",1,"force success")
```

setProcessProgress

Die Funktion "setProcessProgress" legt den Fortschritt eines Prozesses in den folgenden Bereichen fest:

- Operator-Code für Vor- und Nachbearbeitung
- Ein RunJavaScript-Operator

Syntax

```
setProcessProgress (ProcessProgress)
```

Argumente

ProcessProgress

Gibt den Prozentsatzabschluss eines Prozesses an.

Rückgabewert

Es wird kein Wert zurückgegeben (void).

Beispiel

Verwenden Sie die Methode "setProcessProgress", um den Prozessfortschritt folgendermaßen auf 30 Prozent festzulegen:

```
setProcessProgress(30)
```

Wenn ein Prozess den Operator erreicht, zeigt das Dashboard des Betriebsablaufs oder der Prozessdatensatz den Fortschritt als 30 an.

setResourceTotal

Die Funktion "setResourceTotal" legt die Gesamtmenge der Ressourcen mit dem Ressourcennamen "resName" auf den Betrag fest, der im Ressourcenobjekt im Pfad "resPath" festgelegt wurde.

Hinweise:

- Wenn Sie Ressourcen angeben, die nicht als Eingabe vorhanden sind, erstellt diese Funktion die Ressourcen.
- Wenn der Parameter "resName" leer ist, legt diese Funktion die Gesamtmenge auf die Anzahl aller Ressourcen im Ressourcenobjekt fest.

Syntax

```
bSuccess = setResourceTotal(resPath, resName, amount)
```

Argumente

resPath (String)

Gibt den Pfad des Ressourcenobjekts an.

resName (String)

Gibt den Ressourcennamen an, den Sie im Ressourcenobjekt festgelegt haben.

amount (Int)

Gibt die Gesamtmenge an, die Sie für die Ressource festgelegt haben.

Rückgabewert

bSuccess (Boolean)

Gibt "true" zurück, wenn die Funktion erfolgreich ausgeführt wird, oder "false", wenn die Ausführung fehlschlägt.

Beispiel

```
bSuccess1=setResourceTotal(Process.ResObjName, Process.ResName_1,
1);
bSuccess2=setResourceTotal(Process.ResObjName, Process.ResName_3,
3);
```

today

Der Funktion "today" gibt das aktuelle Datum und die aktuelle Uhrzeit zurück. Die zurückgegebene Uhrzeit ist 12:00 AM.

Syntax

```
dtToday = today()
```

Argumente

Keine.

Rückgabewert

dtToday (java.util.Date)

Gibt das aktuelle Datum zurück.

Beispiel

Wenn das heutige Datum 12. Dezember 2009 lautet, wird das Datum 12. Dezember 2009 12:00 AM zurückgegeben.